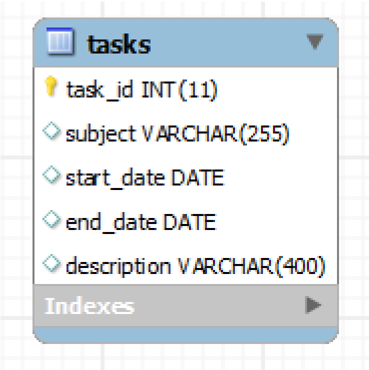
PHP MySQL: Inserting Data Into a Table

Summary: in this lab, you will learn how to use PHP PDO to insert data into a MySQL table.

We will use the tasks table that we created in the create table lab. If you have not yet created the tasks table, please follow the tutorial and create it before going forward with this tutorial.

The following picture illustrates the tasks table:



tasks

task_id INT(11)

subject VARCHAR(255)

start_date DATE

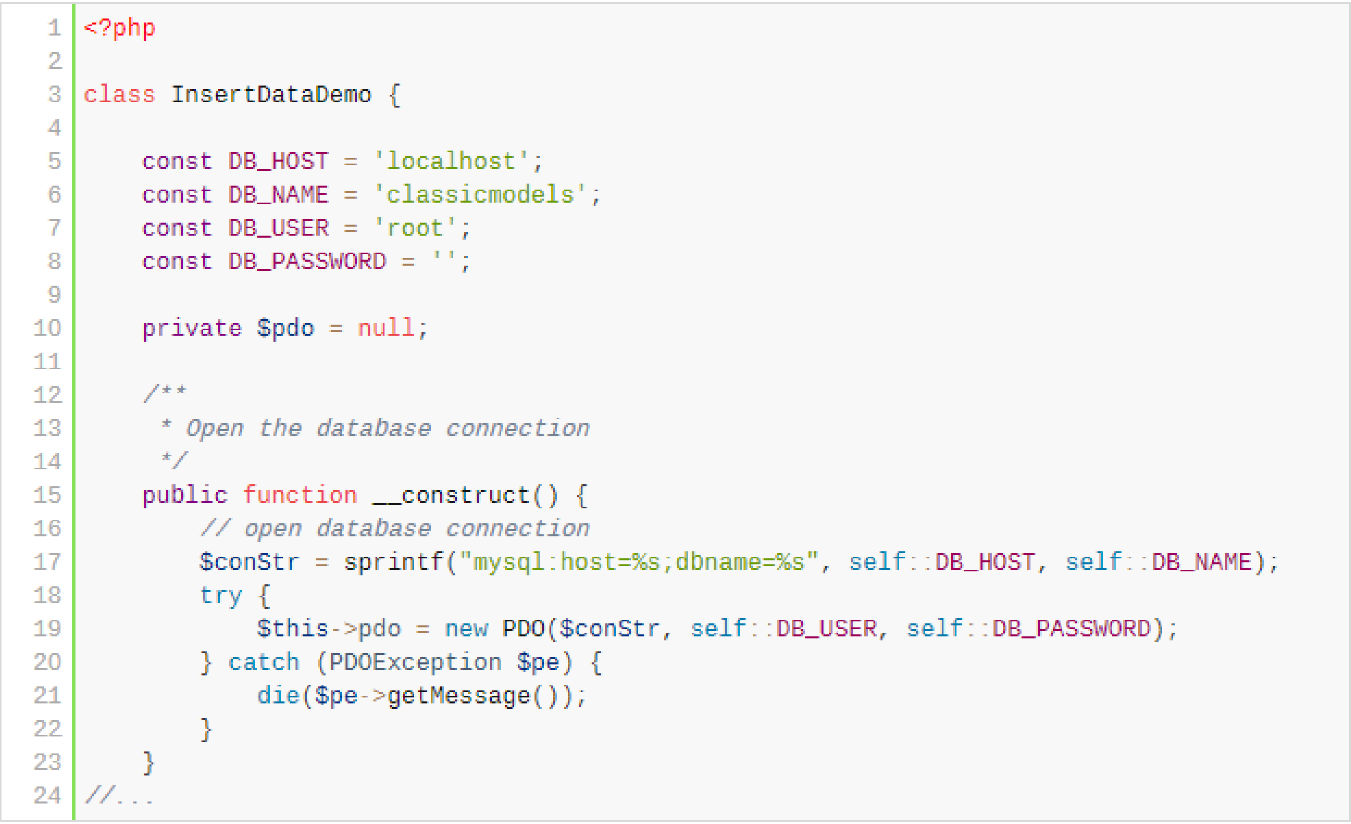
end_date DATE

description VARCHAR(400)

To insert data into a table, you follow the steps below:

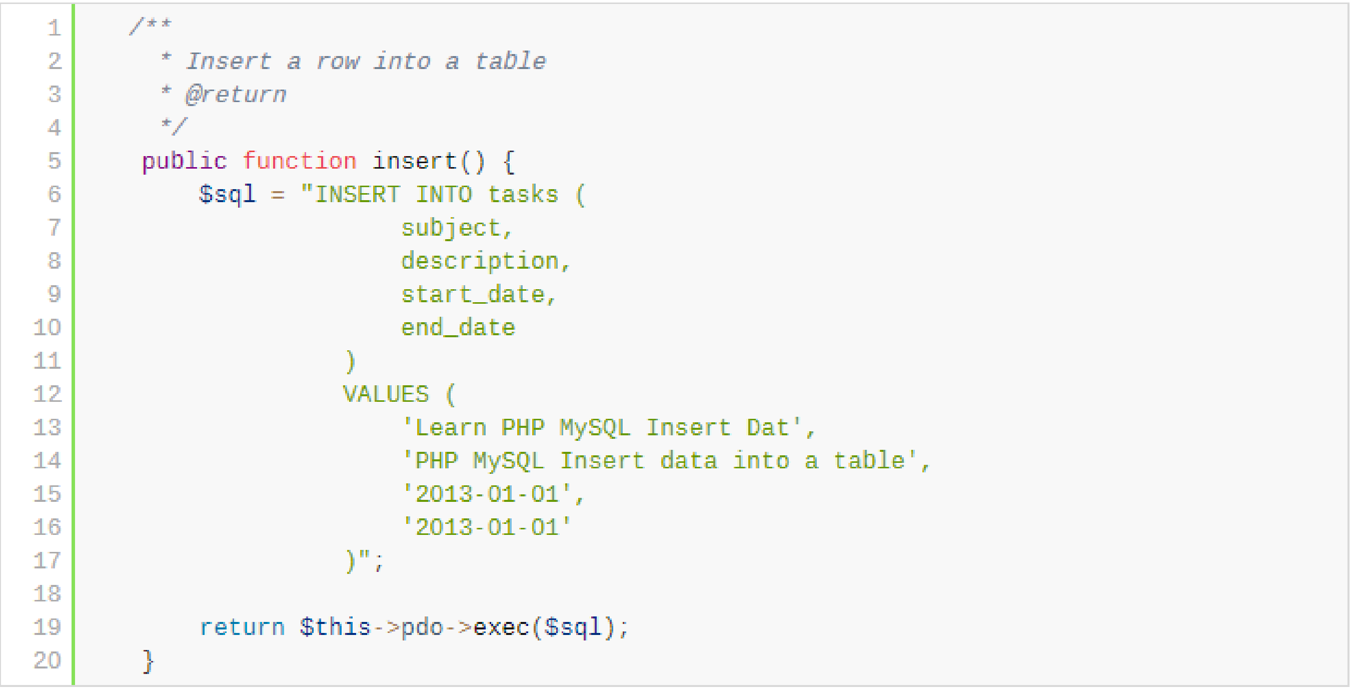
- Connect to the MySQL database by creating a new instance of the PDO object.
- Construct a MySQL INSERT statement.
- Call exec() method of the PDO object.

PHP MySQL: insert a new row into a table exercise



```php
<?php

class InsertDataDemo {

    const DB_HOST = 'localhost';
    const DB_NAME = 'classicmodels';
    const DB_USER = 'root';
    const DB_PASSWORD = '';

    private $pdo = null;

    /**
     * Open the database connection
     */
    public function __construct() {
        // open database connection
        $conStr = sprintf("mysql:host=%s;dbname=%s", self::DB_HOST, self::DB_NAME);
        try {
            $this->pdo = new PDO($conStr, self::DB_USER, self::DB_PASSWORD);
        } catch (PDOException $pe) {
            die($pe->getMessage());
        }
    }
//...
```

The following example illustrates how to insert a new row into the `tasks` table.



```php
   /**
     * Insert a row into a table
     * @return
     */
    public function insert() {
        $sql = "INSERT INTO tasks (
                      subject,
                      description,
                      start_date,
                      end_date
                  )
                  VALUES (
                      'Learn PHP MySQL Insert Dat',
                      'PHP MySQL Insert data into a table',
                      '2013-01-01',
                      '2013-01-01'
                  )";

        return $this->pdo->exec($sql);
    }
```

We defined the `InsertDataDemo` class with a constructor that establishes the database connection and a destructor that closes the database connection. Please refer to the PHP MySQL create table lab for the code for these methods.

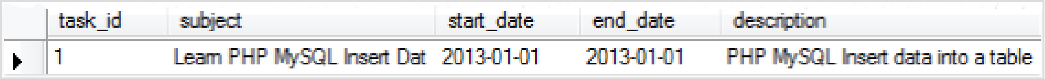
Inside the `InsertDataDemo` class, we define the insert method that calls the `exec()` method of the PDO object to execute the `INSERT` statement.

The following statement creates an instance of the `InsertDataDemo` class and calls the `insert()` method to insert a new row into the tasks table.

```
$obj = new InsertDataDemo();
$obj->insert();
```

Let's query the data in the `tasks` table:

```
SELECT *
  FROM tasks;
```



	task_id	subject	start_date	end_date	description
▶	1	Learn PHP MySQL Insert Dat	2013-01-01	2013-01-01	PHP MySQL Insert data into a table

PHP MySQL: insert a single row using prepared statement exercise

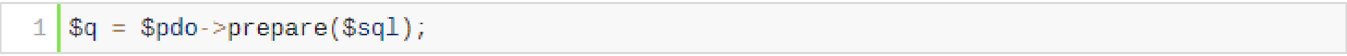
To pass values from PHP to SQL statement dynamically and securely, you use the PDO prepared statement.

First, use MySQL statement with named placeholders as follows:

```
$sql = 'INSERT INTO tasks (
                      subject,
                      description,
                      start_date,
                      end_date
                  )
                  VALUES (
                      :subject,
                      :description,
                      :start_date,
                      :end_date
                  );';
```

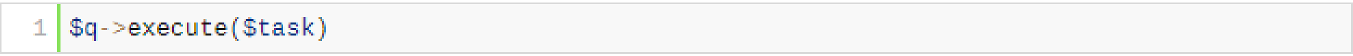
The :subject, :description, :startdate and :enddate are called named placeholders.

Second, call the `prepare()` method of the PDO object to prepare the SQL statement for the execution:



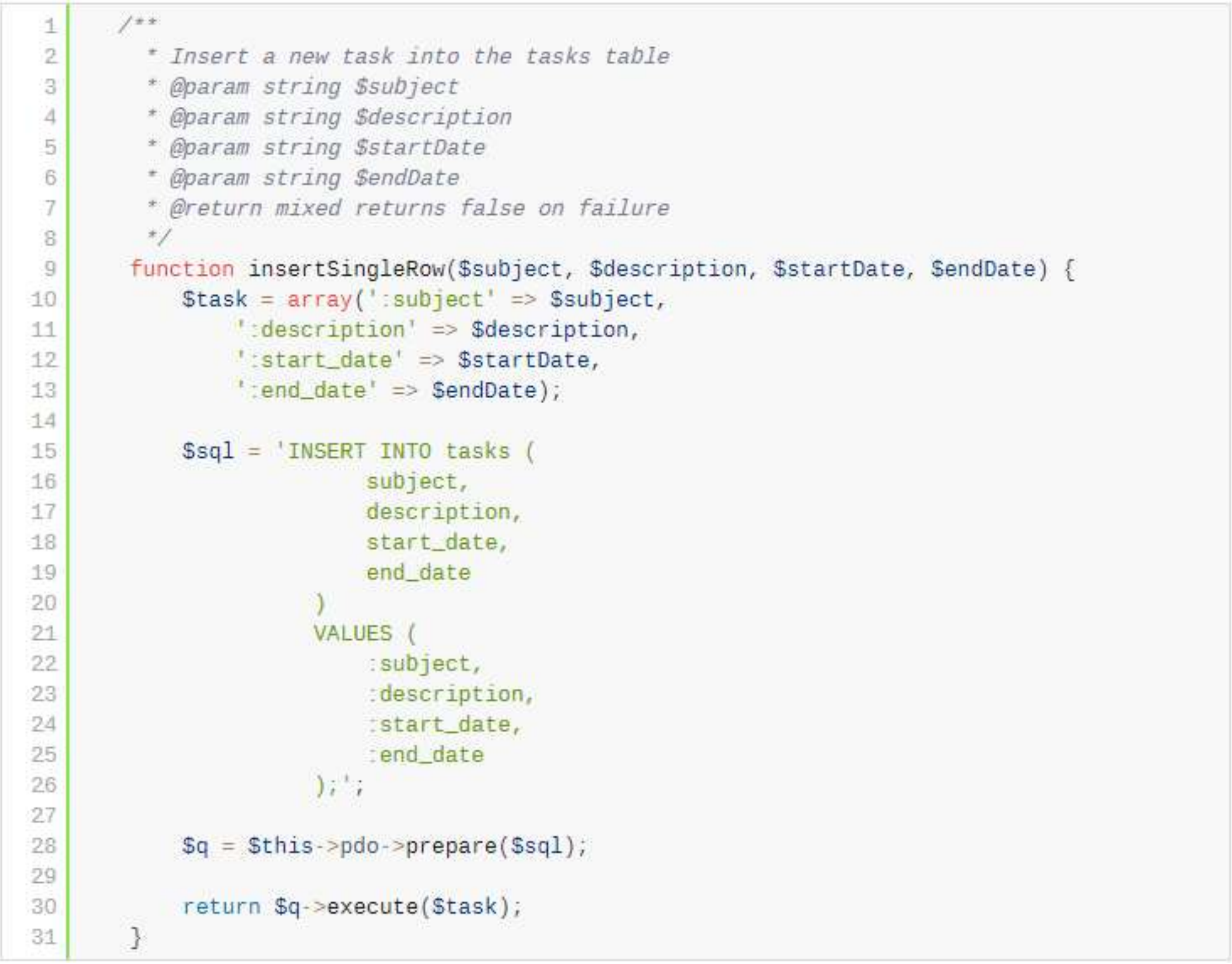
```
$q = $pdo->prepare($sql);
```

Third, call the `execute()` method and pass an array that contains the values which are corresponding to the named placeholders.



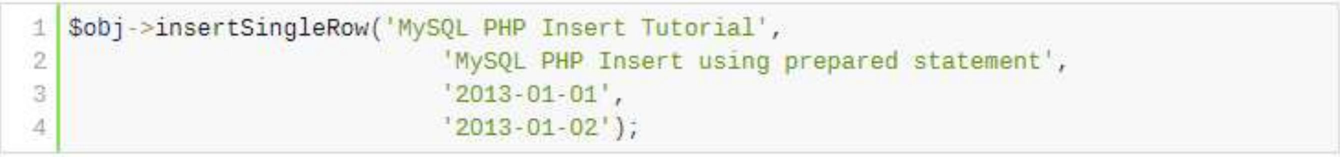
```
$q->execute($task)
```

Putting it all together.



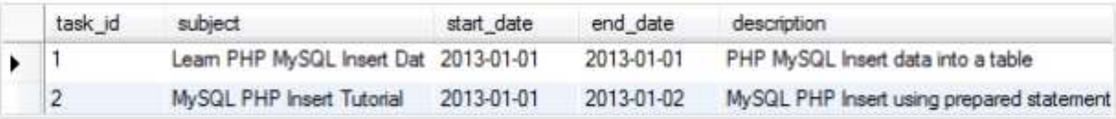
```
/**
  * Insert a new task into the tasks table
  * @param string $subject
  * @param string $description
  * @param string $startDate
  * @param string $endDate
  * @return mixed returns false on failure
  */
function insertSingleRow($subject, $description, $startDate, $endDate) {
    $task = array(':subject' => $subject,
        ':description' => $description,
        ':start_date' => $startDate,
        ':end_date' => $endDate);

    $sql = 'INSERT INTO tasks (
                  subject,
                  description,
                  start_date,
                  end_date
              )
              VALUES (
                  :subject,
                  :description,
                  :start_date,
                  :end_date
              );';

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

    return $q->execute($task);
}
```

Now we can pass the task's data to the `insertSingleRow()` method:



```
$obj->insertSingleRow('MySQL PHP Insert Tutorial',
                            'MySQL PHP Insert using prepared statement',
                            '2013-01-01',
                            '2013-01-02');
```

Le’ts check the `tasks` table:



task_id	subject	start_date	end_date	description
1	Learn PHP MySQL Insert Dat	2013-01-01	2013-01-01	PHP MySQL Insert data into a table
2	MySQL PHP Insert Tutorial	2013-01-01	2013-01-02	MySQL PHP Insert using prepared statement

PHP MySQL Insert multiple rows into a table example

There are two ways to insert multiple rows into a table:

- Execute the `insertSingleRow()` method multiple times.
- Construct a MySQL `INSERT` statement that inserts multiple rows and executes it.