

```
<?php

class DataBase {

    private $db;
    private static $instance = null;

    public function __destruct() {
        $this->db = null;
    }

    private function __construct() {
        $config = Config::getConfig('sqlitedb');
        // SQLite: dsn => 'sqlite:../db/yourdatabase.sqlite/.db', username => '',
password => ''
        // MySQL: dsn => 'mysql:host=localhost:dbname=yourdatabase;charset=utf8mb4',
username => 'root', password => ''
        try {
            $this->db = new PDO($config['dsn'], $config['username'],
$config['password'], [
                PDO::ATTR_PERSISTENT => true,
                PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
                PDO::ATTR_EMULATE_PREPARES => false
            ]);
        } catch (PDOException $e) {
            trigger_error('Could not connect to database: ' . $e->getMessage(),
E_USER_ERROR);
            exit;
        }
    }

    private function getDb() {
        return $this->db;
    }

    public static function getInstance() {
        if (!self::$instance instanceof self) {
            self::$instance = new self;
        }
        return self::$instance->getDb();
    }
}
```

```
57     public static function query() {
58
59         $sql = func_get_arg(0);
60
61         $parameters = array_slice(func_get_args(), 1);
62
63         try {
64             self::getInstance()->beginTransaction();
65
66             $statement = self::getInstance()->prepare($sql);
67
68             $statement->execute($parameters);
69
70             self::getInstance()->commit();
71         } catch (PDOException $e) {
72             self::getInstance()->rollback();
73             // trigger_error('Error: ' . $e->getMessage(), E_USER_ERROR);
74             return false;
75         }
76
77         if ($statement->columnCount() > 0) {
78             return $statement->fetchAll(PDO::FETCH_ASSOC);
79         } else {
80             return $statement->rowCount() == 1;
81         }
82     }
83 }
```