Transactionnel

La méthode beginTransaction() désactive le mode autocommit. Cela signifie que les modifications apportées à la base de données via l'objet PDO ne prendront pas effet tant que vous n'aurez pas appelé la méthode PDO::commit().

```
<?php

// récupérer l’id d’un auteur à partir de son nom et prénom

function get_author_id($pdo, string $first_name, string $last_name)
{
	$sql = 'SELECT author_id
	 FROM authors
	 WHERE first_name = :first_name
	    AND last_name = :last_name';

	$statement = $pdo->prepare($sql);
	$statement->bindParam(':first_name', $first_name, PDO::PARAM_STR);
	$statement->bindParam(':last_name', $last_name, PDO::PARAM_STR);

	if ($statement->execute()) {
		$row = $statement->fetch(PDO::FETCH_ASSOC);
		return $row !== false ? $row['author_id'] : false;
	}

	return false;
}
```

————————————-

```
<?php

function insert_author( $pdo, string $first_name, string $last_name): int
{
	$sql = 'INSERT INTO authors(first_name, last_name)
	 VALUES(:first_name, :last_name)';

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

	$statement->bindParam(':first_name', $first_name, PDO::PARAM_STR);
	$statement->bindParam(':last_name', $last_name, PDO::PARAM_STR);

	$statement->execute();

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

——————————————

```
<?php

function insert_book(\PDO $pdo, string $title, string $isbn, string $published_date, int
$publisher_id): int
{
        $sql = 'INSERT INTO books(title, isbn, published_date, publisher_id)
         VALUES(:title, :isbn, :published_date, :publisher_id)';

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

        $statement->bindParam(':title', $title, PDO::PARAM_STR);
        $statement->bindParam(':isbn', $isbn, PDO::PARAM_STR);
        $statement->bindParam(':published_date', $published_date, PDO::PARAM_STR);
        $statement->bindParam(':publisher_id', $publisher_id, PDO::PARAM_INT);

        $statement->execute();

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

————

```
<?php

function insert_book_author($pdo, int $book_id, int $author_id)
{
        $sql = 'INSERT INTO book_authors(book_id, author_id)
          VALUES(:book_id, :author_id)';

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

        $statement->bindParam(':book_id', $book_id, PDO::PARAM_INT);
        $statement->bindParam(':author_id', $author_id, PDO::PARAM_INT);

        $statement->execute();
}
```

—————-

```
<?php

require 'functions.php';
$pdo = require 'connect.php';

$book = [
        'title' => 'Eternal',
        'isbn' => '9780525539766',
        'published_date' => '2021-03-23',
        'publisher_id' => 2,
];

$author = [
        'first_name' => 'Lisa',
        'last_name' => 'Scottoline',
];

try {
        $pdo->beginTransaction();

        // recherche l’id d’un auteur par le prénom et le nom
        $author_id = get_author_id(
                $pdo,
                $author['first_name'],
                $author['last_name']
        );

        // si l’auteur n’est pas trouvé, création d’un new auteur (INSERT
        if (!$author_id) {
                $author_id = insert_author(
                        $pdo,
                        $author['first_name'],
                        $author['last_name']
                );
        }

        $book_id = insert_book(
                $pdo,
                $book['title'],
                $book['isbn'],
                $book['published_date'],
                $book['publisher_id']
        );

        // insertion sur une table qui fait association

        insert_book_author($pdo, $book_id, $author_id);

        // commit the transaction
        $pdo->commit( );
} catch (PDOException $e) {
        // rollback the transaction
        $pdo->rollBack();

        // show the error message
        die($e->getMessage());
}
```