

Storing Data with Eloquent Models



Mateo Prigl

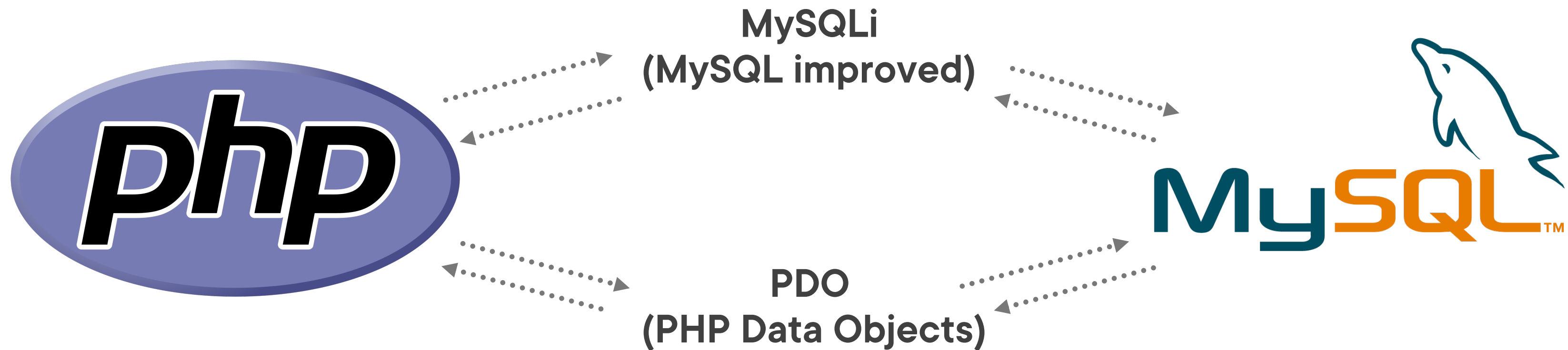
Software developer



Database Management in PHP



Database Management in PHP



Database Query Builder

Database query builder is accessible from the **DB facade**

```
use Illuminate\Support\Facades\DB;

$post = DB::table('posts')->where('title', 'New title')->first();
// SELECT * FROM posts WHERE title = 'New title';

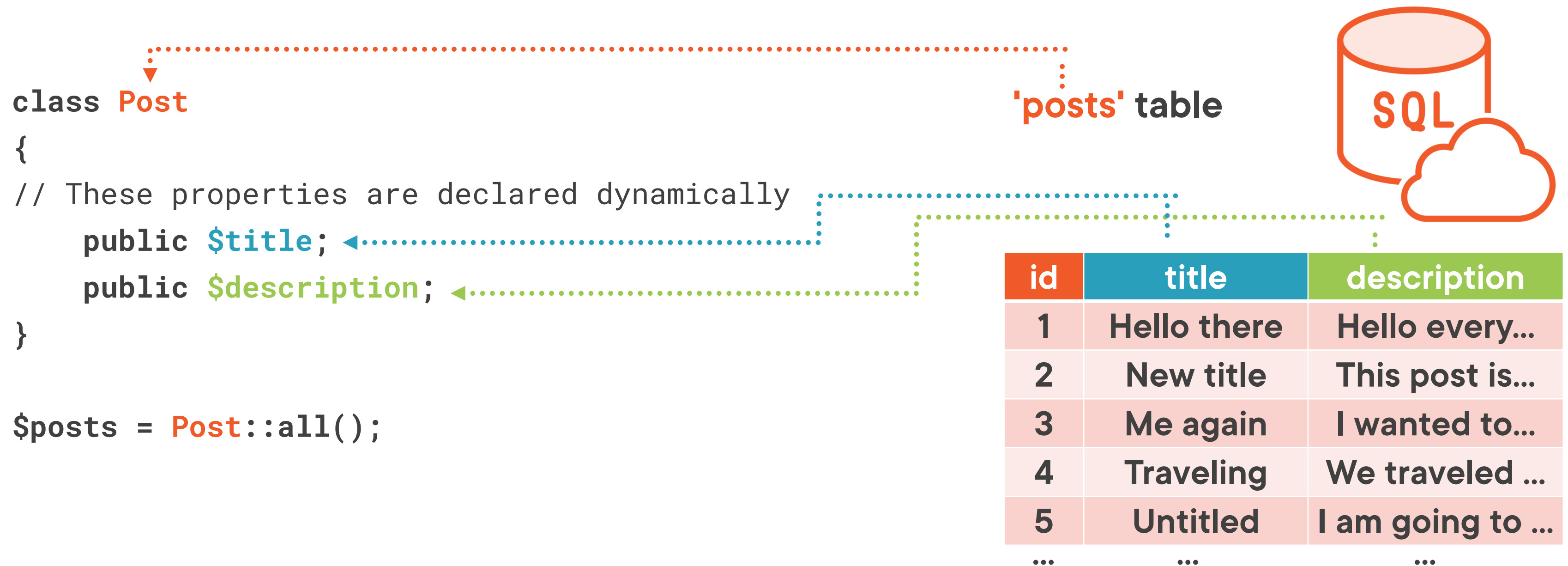
$users = DB::table('users')->get();
// SELECT * FROM 'users';
// Returned object is of the type Illuminate\Support\Collection

foreach ($users as $user) {
    echo $user->email;
}
```



Eloquent ORM

Eloquent is an object-relational mapper (ORM)

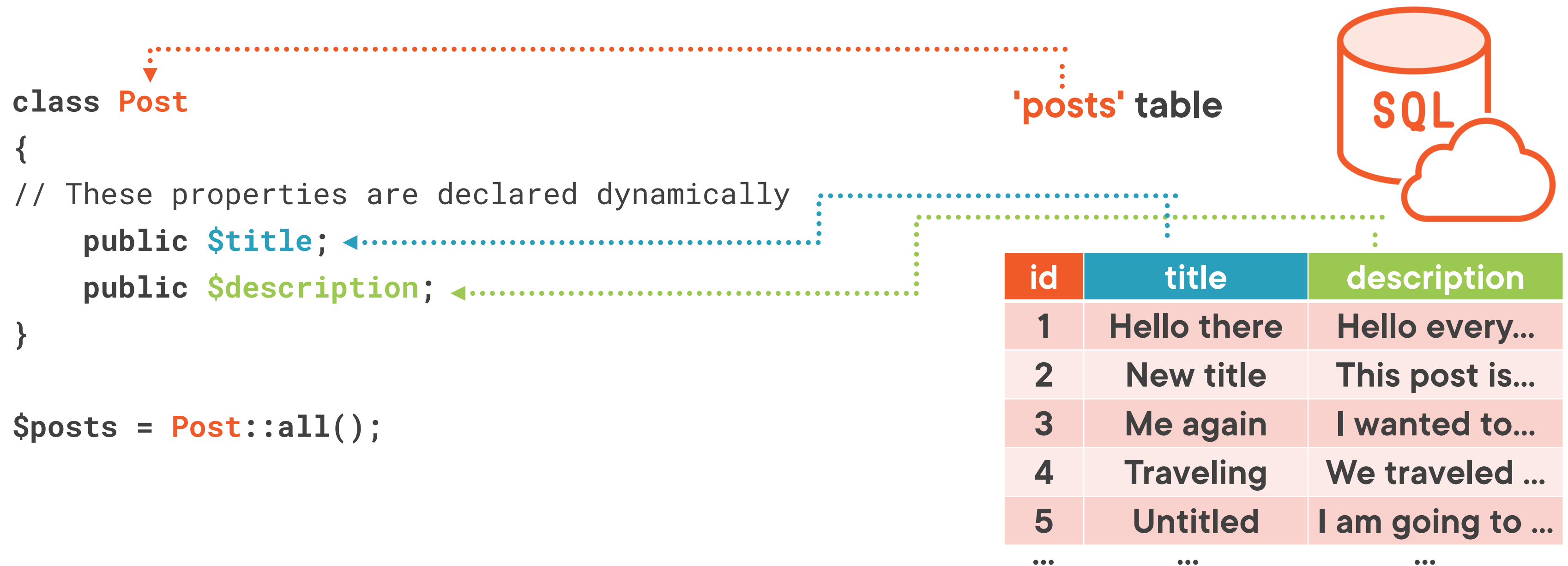


* This is a simplified presentation of the Eloquent ORM functionality



Eloquent ORM

Eloquent is an object-relational mapper (ORM)

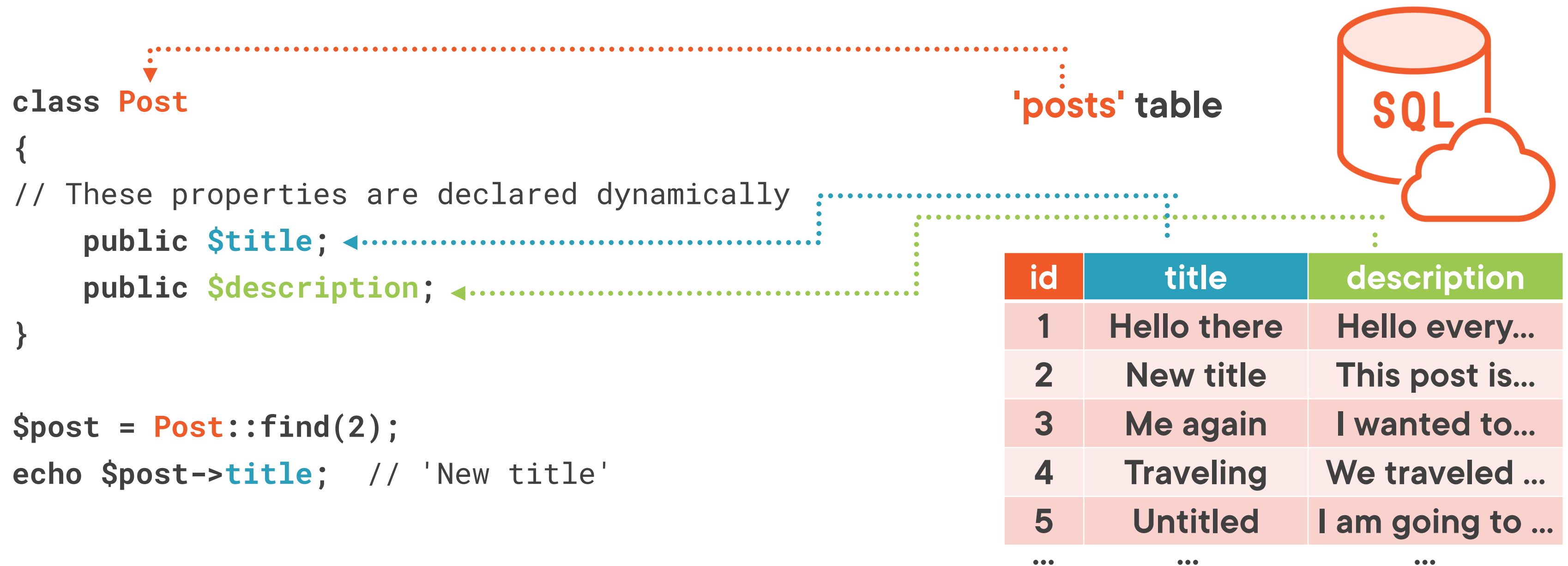


* This is a simplified presentation of the Eloquent ORM functionality



Eloquent ORM

Eloquent is an object-relational mapper (ORM)



* This is a simplified presentation of the Eloquent ORM functionality



Eloquent ORM

Eloquent is an object-relational mapper (ORM)

```
use Illuminate\Database\Eloquent\Model;
```

```
class Post extends Model
```

```
{
```

```
// These properties are declared dynamically
```

```
    public $title;
```

```
    public $description;
```

```
}
```

```
$post = Post::find(2);
```

```
echo $post->title; // 'New title'
```

'posts' table



id	title	description
1	Hello there	Hello every...
2	New title	This post is...
3	Me again	I wanted to...
4	Traveling	We traveled ...
5	Untitled	I am going to ...

* This is a simplified presentation of the Eloquent ORM functionality



Database Migrations

Define the database schema with migration classes

```
public function up()
{
    Schema::create('posts', function (Blueprint $table) {
        $table->id();
        $table->string('title');
        $table->text('description');
    });
}

// CREATE TABLE posts (id BIGINT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY, title VARCHAR(255) NOT
NULL, description TEXT NOT NULL);

public function down()
{
    Schema::drop('posts');
}

// DROP TABLE IF EXISTS posts
```



Why Use DB Facade?

Complex queries

DB has a slightly better performance compared to Eloquent models

Model independence

Work directly with database tables



Retrieve Data with the Post Model

```
use App\Models\Post;

// Retrieves a collection of Post model instances
$posts = Post::where('title', 'new')->orderBy('title')->get();

// Retrieves the first Post model instance
$post = Post::where('title', 'new')->orderBy('title')->first();

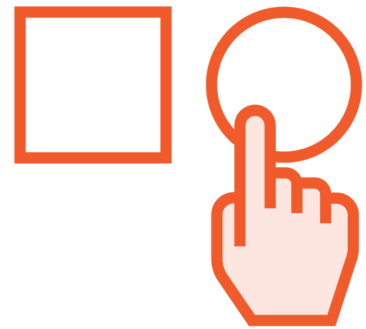
// Retrieves a collection of all Post model instances
$posts = Post::all();
// Returned object is of the type Illuminate\Database\Eloquent\Collection
```



Summary



Connect Laravel to the SQL Database



Choose one of the supported database management systems



Configure the connection in *config/database.php* and *.env*



Create a new database for the Laravel project



Check the Database Connection

Run the command to check if the database is properly configured

```
php artisan db
```

This command will connect you to your database CLI



Database Query Builder

Database query builder is accessible from the **DB facade**

```
use Illuminate\Support\Facades\DB;

$post = DB::table('posts')->where('title', 'New title')->first();
// SELECT * FROM posts WHERE title = 'New title';

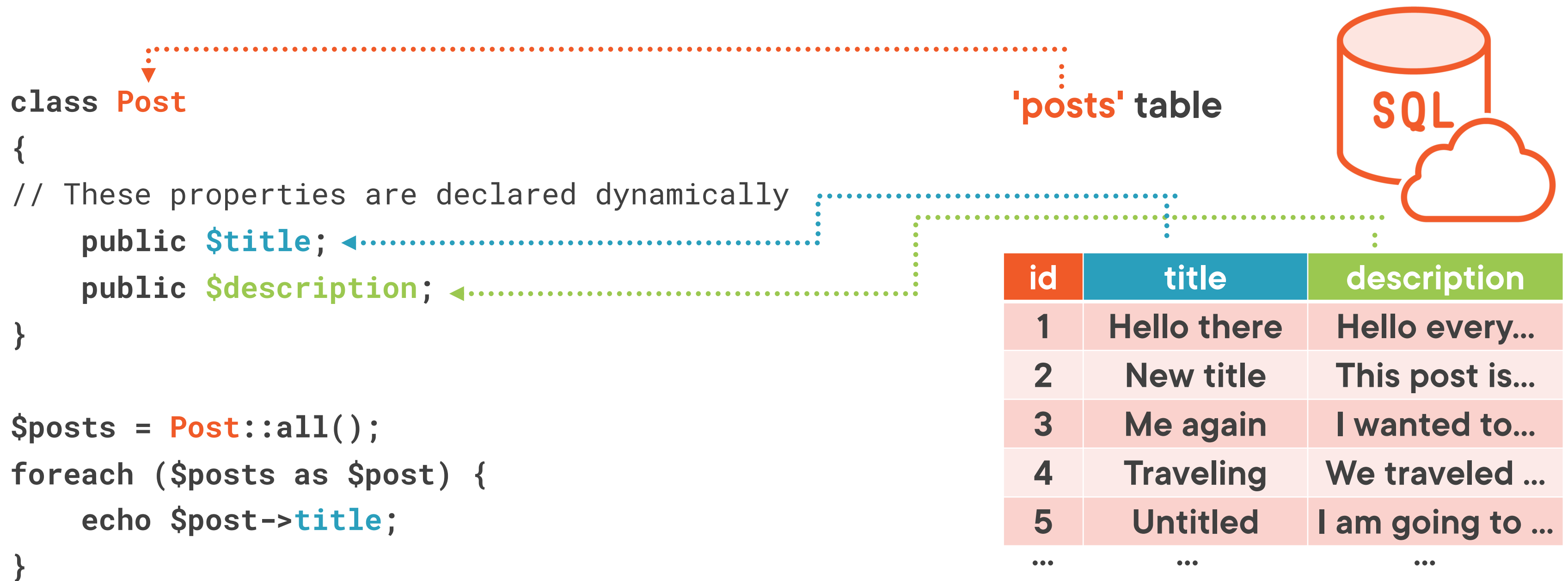
$users = DB::table('users')->get();
// SELECT * FROM 'users';
// Returned object is of the type Illuminate\Support\Collection

foreach ($users as $user) {
    echo $user->email;
}
```



Eloquent ORM

Eloquent is an object-relational mapper (ORM)



* This is a simplified presentation of the Eloquent ORM functionality



Database Migrations

Define the database schema with migration classes

```
public function up()
{
    Schema::create('posts', function (Blueprint $table) {
        $table->id();
        $table->string('title');
        $table->text('description');
    });
}

// CREATE TABLE posts (id BIGINT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY, title VARCHAR(255) NOT
NULL, description TEXT NOT NULL);

public function down()
{
    Schema::drop('posts');
}

// DROP TABLE IF EXISTS posts
```



Create a New Model Class

Commit all of the pending migrations

```
php artisan migrate
```

Create a new model (e.g. *Post*)

```
php artisan make:model Post --migration
```

The *--migration* option will create a corresponding table in the database



Store a New Post Instance in the Database

```
// Instantiate a new Post model object
$post = new Post();

// Assign user input to the appropriate object properties
$post->title = $request->input('title');
$post->description = $request->input('description');

// Commit the changes to the database, save the new post record
$post->save();
```



Retrieve and Show Posts from the Database

Retrieve posts and pass them to the template

```
// Inside of the controller action

// Retrieve all posts from the database
$posts = Post::all();

// Pass the collection to the template
return view('home', ['posts' => $posts]);
```

app/Http/Controllers/HomeController.php

Loop through all of the posts with the *foreach* directive

```
@foreach ($posts as $post)
    {{ $post->title }}
    {{ $post->description }}

@empty
    There are no posts yet.
@endforeach
```

resources/views/home.blade.php



Up Next:

Using Route Parameters to Find a Specific Resource

