

Web Programming Assignment 7

📌 Class	
🕒 Last Edited	@November 16, 2021 11:55 PM
☰ NIM	2301900555
☰ Nama	Alfonsus Ardani Chendrawan

Case

- Migration & Eloquent ORM.** Create simple Laravel project that use relational tables. Use at least 3 tables to describe the relation of One to One relation and One to Many relation (example : Student **have a** Detail personal data, Department **have many** Student, etc).
- Use at least 5 rows data to display. If possible use Seeder technique to fill the tables (find the tutorial for this texhnique), if not use the insertion base on our Session 7 material
 - Capture / screenshot all important code and results (Migrations, Models, Controller, Route, Seeder files if use this technique, Blades and Browser results)

Regards

Initialization

1. Create database in phpMyAdmin called `dbwebproguni`
2. Edit `.env` file:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=dbwebproguni
DB_USERNAME=root
DB_PASSWORD=
```

Model Design/Database Design



1. Create model, seeder, and migration using `php artisan` (in order)

```
php artisan make:model Album -m -s
php artisan make:model AlbumStatistic -m -s
php artisan make:model Song -m -s
```

2. Create controller using the `php artisan` for this whole session. (Does not follow MVC paradigm, for simplicity purposes)

```
php artisan make:controller Session7Controller
```

Migration Configuration

- Albums Table

database/migrations/2021_11_14_082118_create_albums_table.php

```
<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
class CreateAlbumsTable extends Migration
{
    public function up()
    {
        Schema::create('albums', function (Blueprint $table) {
            $table->id();
            $table->string('title');
            $table->string('genre');
            $table->integer('year');
            $table->timestamps();
        });
    }
    public function down()
    {
        Schema::dropIfExists('albums');
    }
}
```

- Album Statistics Table

database/migrations/2021_11_14_082241_create_album_statistics_table.php

```
<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
class CreateAlbumStatisticsTable extends Migration
{
    public function up()
    {
        Schema::create('album_statistics', function (Blueprint $table) {
            $table->id();
            $table->foreignId('album_id');
            $table->foreign('album_id')->references('id')->on('albums');
            $table->integer('have');
            $table->integer('want');
            $table->float('rating');
            $table->integer('ratings');
            $table->timestamps();
        });
    }
    public function down()
    {
        Schema::dropIfExists('album_statistics');
    }
}
```

- Songs Table

database/migrations/2021_11_14_082255_create_songs_table.php

```

<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
class CreateSongsTable extends Migration
{
    public function up()
    {
        Schema::create('songs', function (Blueprint $table) {
            $table->id();
            $table->string('title');
            $table->string('author');
            $table->foreignId('album_id');
            $table->foreign('album_id')->references('id')->on('albums');
            $table->timestamps();
        });
    }
    public function down()
    {
        Schema::dropIfExists('songs');
    }
}

```

Seeders Configuration

- Albums

```

<?php
namespace Database\Seeders;
use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;
class AlbumSeeder extends Seeder
{
    public function run()
    {
        DB::table('albums')->insert([
            [
                'title' => 'Worlds',
                'genre' => 'Electronic',
                'year' => 2014,
            ],
            [
                'title' => 'Nurture',
                'genre' => 'Electronic, Pop, Classical',
                'year' => 2021,
            ],
        ]);
    }
}

```

- Album Statistics

```

<?php
namespace Database\Seeders;
use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;
class AlbumStatisticSeeder extends Seeder
{
    public function run()
    {
        DB::table('album_statistics')->insert([
            [
                'album_id' => 1,
                'have' => 4336,
                'want' => 2905,
                'rating' => 4.51,
                'ratings' => 538,
            ],
            [
                'album_id' => 2,
                'have' => 1917,
                'want' => 629,
                'rating' => 4.53,
                'ratings' => 201,
            ],
        ]);
    }
}

```

```
}  
}
```

- Songs

```
<?php  
namespace Database\Seeders;  
use Illuminate\Database\Seeder;  
use Illuminate\Support\Facades\DB;  
class SongSeeder extends Seeder  
{  
    public function run()  
    {  
        DB::table('songs')->insert([  
            ['title' => 'Divinity','author' => 'Porter Robinson','album_id' => 1],  
            ['title' => 'Sad Machine','author' => 'Porter Robinson','album_id' => 1],  
            ['title' => 'Flicker','author' => 'Porter Robinson','album_id' => 1],  
            ['title' => 'Fresh Static Snow ', 'author' => 'Porter Robinson','album_id' => 1],  
            ['title' => 'Wind Tempos','author' => 'Porter Robinson','album_id' => 2],  
            ['title' => 'Do-re-mi-fa-so-la-ti-do','author' => 'Porter Robinson','album_id' => 2],  
            ['title' => 'Mirror','author' => 'Porter Robinson','album_id' => 2],  
        ]);  
    }  
}
```

Routing Configuration

```
<?php  
use Illuminate\Support\Facades\Route;  
use App\Http\Controllers\Session7Controller;  
Route::get('/song-list', [Session7Controller::class, 'viewSongList']);
```

Controller Configuration

```
public function viewSongList(){  
    $albums = Album::all();  
    return view('songs',['albums'=>$albums]);  
}
```

Model Configuration with **HasFactory**

Reference:



- Album

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
class Album extends Model
{
    use HasFactory;
    public function song(){
        return $this->hasMany(Song::class);
    }
    public function stat(){
        return $this->hasOne(AlbumStatistic::class);
    }
}
```

- Album Statistics

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
class AlbumStatistic extends Model
{
    use HasFactory;
    public function album(){
        return $this->belongsTo(Album::class);
    }
}
```

- Song

```
<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
class Song extends Model
{
    use HasFactory;
    public function album(){
        return $this->belongsTo(Album::class);
    }
}
```

Blade Template View Configuration

- songs.blade.php

```
<body style="font-family:sans-serif;background-color: rgb(239, 241, 252);">
<div style="margin-top:100px;"><h1 style="color:rgb(134, 108, 206); text-align: center; font-size: 5em">Song List</h1></div>
<div style="display:flex;">
@foreach ($albums as $album)
    <div style="border-radius:20px; color:rgb(53, 53, 53); background-color: rgb(255, 255, 255);padding:30px;margin:20px;width:20%;height:100%">
        <h2>{{ $album->title }}</h2>
        <div><b>{{ $album->year }}</b></div>
        <div>{{ $album->genre }}</div>
        <br>
        <div>
            @foreach ($album->song as $song)
                <div style="color:white; border-radius:15px; background-color: rgb(134, 108, 206);padding:20px;margin-bottom:10px;">
                    <b>{{ $song->title }}</b> by {{ $song->author }}
                </div>
            @endforeach
        </div>
        <h3>Statistics</h3>
        <div style="display:flex; flex-direction: row; justify-content:space-between">
            <div style="display:flex; flex-direction: column; width:50%;">
                <div> Have: <b>{{ $song->album->stat->have }}</b></div>
                <div> Want: <b>{{ $song->album->stat->want }}</b></div>
            </div>
            <div style="display:flex; flex-direction: column; width:50%;">
                <div> Rating: <b>{{ $song->album->stat->rating }}</b></div>
                <div> Ratings: <b>{{ $song->album->stat->ratings }}</b></div>
            </div>
        </div>
    </div>
@endforeach
</div>
</body>
```

Browser View

