Example:1

<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;

use Illuminate\Database\Seeder;

use Illuminate\Support\Facades\DB;

class StudentSeeder extends Seeder

{

    use WithoutModelEvents;

    /\*\*

     \* Run the database seeds.

     \*/

    public function run(): void

    {

        $students = [];

        $count  = 0;

        for ($i = 1; $i <= 1000000; $i++) {

            $count++;

            $students[] = [

                'student\_number' => str\_pad((string) $i, 4, '0', STR\_PAD\_LEFT),

                'first\_name' => fake()->firstName(),

                'last\_name' => fake()->lastName(),

                'middle\_name' => fake()->firstName(),

                'nstp\_fee' => fake()->randomElement([700,1500]),

                'created\_at' => now(),

                'updated\_at' => now(),

            ];

            if($count % 1000 == 0){

                echo $count . ' records inserted' . PHP\_EOL;

                DB::table('students')->insert($students);

                $students = [];

                $count = 0;

                echo 'Bulk insert done' . PHP\_EOL;

                sleep(1);

            }

        }

    }

}

Example 2

<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;

use App\Enum\Enrollment\StudentEnrollmentTypeEnum;

use App\Models\Program;

use App\Models\SchoolYear;

use App\Models\Section;

use App\Models\SemesterFee;

use App\Models\Student;

use App\Models\YearLevel;

use Illuminate\Support\Facades\DB;

class CustomEnrollmentSeeder extends Seeder

{

    protected static $students;

    protected static $semesterFees;

    protected static $yearLevels;

    protected static $programs;

    protected static $sections;

    protected static $schoolYears;

    public function run(): void

    {

        // preload all records into memory (avoids 500k queries)

        self::$students     ??= Student::all();

        self::$semesterFees ??= SemesterFee::all();

        self::$yearLevels   ??= YearLevel::all();

        self::$programs     ??= Program::all();

        self::$sections     ??= Section::all();

        self::$schoolYears  ??= SchoolYear::all();

        $enrollments = [];

        $count = 0;

        for ($i = 1; $i <= 500000; $i++) {

            // pick random related models

            $student     = self::$students->random();

            $semesterFee = self::$semesterFees->random();

            $yearLevel   = self::$yearLevels->random();

            $program     = self::$programs->random();

            $section     = self::$sections->random();

            $schoolYear  = self::$schoolYears->random();

            $enrollType  = collect(StudentEnrollmentTypeEnum::cases())->random()->value;

            $enrollments[] = [

                'semester\_fee\_id' => $semesterFee->id,

                'student\_id'      => $student->id,

                'student\_number'  => $student->student\_number,

                'student\_first\_name' => $student->first\_name,

                'student\_last\_name'  => $student->last\_name,

                'student\_middle\_name'=> $student->middle\_name,

                'semester\_name'      => $semesterFee->name,

                'year\_level\_name'    => $yearLevel->name,

                'program\_name'       => $program->name,

                'program\_type'       => $program->type,

                'section\_name'       => $section->name,

                'student\_enrollment\_type' => $enrollType,

                'is\_regular\_student' => fake()->boolean(),

                'registration\_fee'            => fake()->randomFloat(2, 0, 10000),

                'library\_fee'                 => fake()->randomFloat(2, 0, 10000),

                'guidance\_fee'                => fake()->randomFloat(2, 0, 10000),

                'internet\_fee'                => fake()->randomFloat(2, 0, 10000),

                'typing\_fee'                  => fake()->randomFloat(2, 0, 10000),

                'aircon\_fee'                  => fake()->randomFloat(2, 0, 10000),

                'other\_fee'                   => fake()->randomFloat(2, 0, 10000),

                'nstp\_fee'                    => fake()->randomFloat(2, 0, 10000),

                'rate\_per\_unit'               => fake()->randomFloat(2, 0, 10000),

                'rate\_per\_laboratory\_subject' => fake()->randomFloat(2, 0, 10000),

                'rate\_thesis\_fee'             => fake()->randomFloat(2, 0, 10000),

                'tuition\_fee'                 => fake()->randomFloat(2, 0, 10000),

                'discount'                    => fake()->randomFloat(2, 0, 10000),

                'down\_payment'                => fake()->randomFloat(2, 0, 10000),

                'school\_year' => $schoolYear->start\_year . '-' . $schoolYear->end\_year,

            ];

            $count++;

            // bulk insert every 1000 records

            if ($count % 1000 == 0) {

                DB::table('enrollments')->insert($enrollments);

                echo $count . " records inserted\n";

                echo "Bulk insert done\n";

                $enrollments = [];

                $count = 0;

                sleep(1); // optional

            }

        }

        // insert any leftover records

        if (!empty($enrollments)) {

            DB::table('enrollments')->insert($enrollments);

            echo "Final bulk insert done\n";

        }

    }

}