**TABLE NAMING ERROR**

**Связь «Один-со-Многими» и «Один-с-Одним» (6.1.4.1.)**

/\*

По соглашению именования в Laravel, имя таблицы должно быть во множественном числе (car -> cars). Здесь допущена орфографическая ошибка: первичные = primar**ie**s, вторичные = secondar**ie**s. Поэтому связи должны быть указанны явно.

\*/

**Миграция первичной таблицы:**

Schema::create('primarys', function (Blueprint $table) {

$table->id();

$table->string('userName', 50);

$table->timestamps();

});

**Миграция вторичной таблицы:**

Schema::create('secondarys', function (Blueprint $table) {

$table->id();

$table->string('car', 50);

$table->foreignId('primary\_id')->constrained('primarys', 'id')->onDelete('cascade');

$table->timestamps();

});

/\*

foreignId('primary\_id') - создать во вторичной таблице поле с именем 'primary\_id';

constrained('primarys', 'id') - методу constrained(), при выполнении соглашения, ничего передавать не нужно, он "сам возмет из" foreignId('primary\_id') имя первичной таблицы "primary" и столбец с которым нужно создать связь "id", **НО** имя таблицы будет преобразованно во множественное число - "primar**ie**s" и не будет совпадать с нашим именем таблицы "primar**y**s". Поэтому методу явно передается имя таблицы и колонки.

\*/

**Модель первичной таблицы:**

class Primary extends Model{

protected $table = 'primarys';

protected $fillable = ['userName'];

public function secondary(){

return $this->hasMany(Secondary::class);

}

}

**Модель вторичной таблицы:**

class Secondary extends Model{

protected $table = 'secondarys';

protected $fillable = ['car'];

/\*

Внимание! Имя метода связи вторичной модели с первичной должно совпадать с именем колонки вторичной таблице по таким правилам: **primary**() = **primary**\_id

\*/

public function primary(){

return $this->belongsTo(Primary::class);

}

}

/\*

В моделях Primary и Secondary свойство protected $table явно указывает имя таблицы с которой взаимодействует модель, поскольку не выполняется соглашение по именованию таблиц Laravel. Имя класса Primary преобразовывается в имя таблицы "primar**ie**s", которая в деуствительности называется "primar**y**s"

\*/

**Artisan tinker:**

use App\Models\Primary as P;

use App\Models\Secondary as S;

//Создать объект первичной модели и сохранить в БД:

$objP = new P();

$objP->userName = 'Anton';

$objP->save();

//Создаем объект первичной модели из первого кортежа первичной таблицы:

$objP = P::find(1);

$objS = new S(); - создали пустой объект вторичной модели

$objS->car = 'bmw'; - записали новое значение в поле car

//**Сохраняем запись** во вторичную таблицу связывая ее с первичной таблицей **при помощи вторичной модели** и ее метода связи с первичной моделью primary():

$objS->primary()->associate($objP);

$objS->save();

$objS2 = new S(); - создаем еще один объект(второй) вторичной модели

$objS2->car = 'Lotus'; - записали новое значение в поле car

//**Сохраняем запись** во вторичную таблицу связывая ее с первичной таблицей **при помощи первичной модели** и ее метода связи со вторичной моделью secondary():

$objP->secondary()->save($objS2);