ONE TO MANY

Dobbiamo effettuare una ***Foreign Key Migration***.
Aggiungiamo la colonna *ID* della tabella *User* alla tavella *Travel*.

Nella *bash*:

```
php artisan make:migration add_user_id_column_to_travels_table
```

Nel *migration*:

```
public function up(){
	Schema::table('travel', function (Blueprint $table){
		$table->unsignedBigInteger('iser_id')->after('img')->default(1); //1
		$table->foreign('user_id')->references('id')->on('users') //2
});
}


public function down(){
	Schema::table('travel', function (Blueprint $table){
		$table->dropForeign(['user_id']); //2
		$table->dropColumn('user_id'); //1
	});
}
```

unsignedBigInteger . Indichiamo che il tipo di dato e' un numero integrale non auto incrementale
default(1) . Indichiamo un valore di default per i valori precedenti alla creazione non ancora collegati.
after('img') . Scegliere dove posizionare la colonna. In questo caso dopo la colonna img.
//1 . La prima istruzione crea la colonna
//2 . La seconda istruzione crea la relazione. Indica che la colonna *user_id* e' una chiave esterna che si riferisce alla colonna *id* della tabella *users*

Nella *bash* effettuiamo la ***migrazione***:

```
php artisan migrate
```

Ora dobbiamo mettere in ***relazione i Model***.

Nel Model *User* aggiungiamo la funzione:

```
public function travels(){
	return $this->hasMany(Travel::class);
}
```

hasMany() . Lo *user $this* ha molti viaggi. Stiamo dichiarando che e' una rotta One to Many.

Nel Model *Travel*:

```
public function user(){
	return $this->belongsTo(User::class);
}
```

belongsTo . Il Model travel appartiene soltanto ad un utente.

Dobbiamo adesso richiamare il nome dell'utente Loggato sul sito.
Nella navbar troviamo la funzione ***Auth::user()->name*** che sarà la chiave.

NB: Istruiamo il Travel *Model* aggiungendo tra i ***$fillable*** '*user_id*'.

Successivamente istruiamo il ***Mass Assignment***:

```
public function store(Request $request){

// 1. Metodo Avanzato
	$user = Auth::user();
	$user->travels()->create([
		'title'=>$request->title,
		'img'=>$request->file('img')->store('public/img')
	]);

// 2. Metodo Semplice
	$travel = Travel::create([
		'title'=>$request->title,
		'img'=>$request->file('img')->store('public/img'),
		'user_id'=>Auth::id()
	]);
	return redirect(route('travel.index'));
}
```

*//***1** . *Metodo avanzato.* ***Richiamiamo la relazione*** tra i due modelli per creare un oggetto di classe Travel dal modello di classe User, grazie alla relazione che abbiamo specificato con la funzione travels(). E' una specie di ***relazione implicita***.
*//***2** . *Metodo semplice*. Diciamo ad *user_id* di avere come valore l'ID dell'utente Loggato.

Nell'**HTML** adesso possiamo inserire il nome dell'utente relazionato al post che verrà creato.

```
{{$travel->user->name}}
```

user . Attraverso $travel prendiamo l'utente a cui e' associato quel viaggio. Quell'*user* e' la funzione *user()* nel *Model User.*

Oppure richiamare tutta la collection di un determinato utente.

```
@foreach (Auth::user()->travels as $user_travel)
    {{$user_travel->title}}
@endforeach
```