

Reference key Readme

In this demo, we are going to use a reference_id as foreign key instead of id(primary key) as foreign key in other table.

Here we have 2 tables(described below with columns) :

Users :

id(primary key), name, email, password, user_id(reference key)

Profile:

id(primary key), profile_id(reference_key), user_id(foreign key)

To get the data into user table, let's add some fake values.

```
>>> $user = \App\Models\User::create(['name'=>'sandeep','email'=>'sandy@yopmail.com','password'=>'abc123','user_id'=>1]);
=> App\Models\User {#2904
    name: "sandeep",
    email: "sandy@yopmail.com",
    user_id: 1,
    updated_at: "2018-11-14 09:55:38",
    created_at: "2018-11-14 09:55:38",
    id: 1,
}
>>> $user = \App\Models\User::create(['name'=>'constellation','email'=>'constellation@yopmail.com','password'=>'abc123','user_id'=>2]);
=> App\Models\User {#2915
    name: "constellation",
    email: "constellation@yopmail.com",
    user_id: 2,
    updated_at: "2018-11-14 09:57:05",
    created_at: "2018-11-14 09:57:05",
    id: 2,
}
>>> 
```

We also add data to profile table :

```
>>> $profile = \App\Models\Profile::create(['profile_id'=>1,'user_id'=>1]);
=> App\Models\Profile {#2896
    profile_id: 1,
    user_id: 1,
    updated_at: "2018-11-14 09:59:00",
    created_at: "2018-11-14 09:59:00",
    id: 1,
}
>>> $profile = \App\Models\Profile::create(['profile_id'=>2,'user_id'=>2]);
=> App\Models\Profile {#2904
    profile_id: 2,
    user_id: 2,
    updated_at: "2018-11-14 09:59:17",
    created_at: "2018-11-14 09:59:17",
    id: 2,
}
>>>
```

Now since we have both user and profile tables linked with reference is column : user_id, here we get the user data along with the profile information.

```
ucrate@ucrate-Latitude-3480 /var/www/html/php tea talks/POCs/keys > master
php artisan tinker
Psy Shell v0.9.9 (PHP 7.1.20-1+ubuntu16.04.1+deb.sury.org+1 - cli) by Justin Hil
eman
>>> $user = \App\Models\User::with('profile')->get();
=> Illuminate\Database\Eloquent\Collection {#2896
    all: [
        App\Models\User {#2903
            id: 1,
            name: "sandeep",
            email: "sandy@yopmail.com",
            email_verified_at: null,
            created_at: "2018-11-14 09:55:38",
            updated_at: "2018-11-14 09:55:38",
            user_id: 1,
            profile: App\Models\Profile {#2909
                id: 1,
                profile_id: 1,
                user_id: 1,
                created_at: "2018-11-14 09:59:00",
                updated_at: "2018-11-14 09:59:00",
            },
        },
        App\Models\User {#2904
            id: 2,
            name: "constellation",
            email: "constellation@yopmail.com",
            email_verified_at: null,
            created_at: "2018-11-14 09:57:05",
            updated_at: "2018-11-14 09:57:05",
            user_id: 2,
            profile: App\Models\Profile {#2910
                id: 2,
                profile_id: 2,
                user_id: 2,
                created_at: "2018-11-14 09:59:17",
                updated_at: "2018-11-14 09:59:17",
            },
        },
    ],
}
>>>
```

We get this data as we have defined profile relation in user model as :

```
return $this->hasOne('App\Models\Profile', 'user_id', 'user_id');
```

The user_id of profile to user_id of user table!!
