**ابزارهای Eager Loading و Lazy Loading برای دریافت داده های مرتبط مدلها ( مدلهایی که با هم رابطه دارند) از دیتابیس استفاده میشن.**

**منظور از داده های مرتبط چیه؟ دیتاهایی که با دیتاهای اصلی در دیتابیس در ارتباط هستند و معمولا از طریق انواع روابط؛ به هم متصلن.**

**مثال:**

**کاربر و پروفایل ==> هر کاربر یک پروفایل داره و هر پروفایل؛ مطعلق به یک کاربره ==> پروفایل؛ داده ی مرتبط به کاربر هستش.**

**پست ها و کامنت ها ==> هر پست؛ چندین کامنت داره ==> کامنت ها؛ داده های مرتبط با پست ها هستند.**

**دانش آموزان و کلاس ها ==> هر کلاس؛ چندین دانش آموز داره؛ هر دانش آموز تو چندین کلاس شرکت میکنه ==> دانش آموزان و کلاس ها؛ داده های مرتبط هستند**

**با استفاده از Eager Loading و Lazy Loading میتونیم؛ داده‌های مرتبط رو؛ بصورت بهینه تری دریافت کنیم.**

**### مشکل N+1 Query**

**مشکل N+1 Query زمانی رخ میده که ما میخوایم داده های یک مدل + داده های مرتبط به اون مدل رو از دیتابیس بگیریم. مثال: داده های مدل User و داده های مرتبط مدل Post.**

**مثال: مدل User با مدل Post رابطه داره و میخوایم تمام یوزرها و پست هاشون رو بگیریم.**

**اگه از Lazy Loading استفاده کنیم به ازای هر کاربر یک کوئری جدا برای دریافت پست‌های اون کاربر باید بزنیم**

**اگه 10 تا کاربر داشته باشی؛ یک کوئری واسه دریافت کاربران نیازه + 10 تا کوئری جدا واسه گرفتن پست های هر کاربر ==> جمعا 11 تا کوئری به دیتابیس باید زده بشه که باعث افزایش بار**

**روی سرور و کاهش کارایی می‌شود.**

**ولی اگه از Eager Loading استفاده بشه؛ تمام دیتاهای بالا رو با یکی 2تا کوئری میتونیم بگیریم.**

**Eager Loading: این ابزار؛ دیتاهای اصلی و دیتاهای مرتبط یک مدل رو؛ فقط با یک کوئری؛ از دیتابیس دریافت میکنه. ==> با استفاده از متد with() در Eloquent**

**پس وقتی که رکوردهای مربوط به یک مدل رو؛ از دیتابیس میگیریم، تمامی داده‌های مرتبط با آن مدل، به صورت خودکار و در همان لحظه دریافت میشن.**

**مثال: وقتی رکورد مربوط به یک یوزر رو از دیتابیس میگیری؛ تمام پست هاش هم گرفته میشه**

**زمانی از Eager Loading استفاده میکنیم که به داده های مرتبط؛ نیاز داریم و میخوایم از مشکل N+1 Query جلوگیری کنیم**

**این روش تعداد کوئری ها به دیتابیس رو کم میکنه و عملکرد – پرفورمنس برنامه رو خیلی بهتر میکنه**

**دریافت داده های مرتبط؛ از دیتابیس؛ بصورت بهینه تری انجام بشه.**

**این دو ابزار کمک میکنن تعداد کوئری ها به دیتابیس کمتر بشه و در نتیجه؛ عملکرد نرم افزارمون رو بهبود بدیم.**

**مثال: $posts = Post::with('comments')->get();**

**پس؛ استفاده از eager loading باعث میشه تعداد کوئری ها کمتر بشه (یک یا دو کوئری) ==> بهینه سازی**

**در نتیجه پرفورمنس نرم افزار بهتر میشه و زمان پاسخگویی برنامه کمتر بشه.**

**وقتی میگیم تعداد کوئری‌های دیتابیس را “بهینه” کنیم، منظور این که:**

**تعداد کوئری‌ها را به گونه‌ای کاهش بدیم که کمتر از منابع سیستم استفاده بشه (استفاده ی بهینه – کارآمد) + با کمترین تعداد کوئری ممکن، بیشترین داده‌های مورد نیاز را دریافت کنیم و**

**عملکرد برنامه بهتر بشه. ==> مثال: با lazy loading کلی کوئری باید بزنی تا کامنت های پست ها رو دریافت کنی؛ ولی با eager loading با یکی دوتا کوئری؛ همون نتیجه رو میتونی بگیری.**

**مزایای بهینه‌سازی کوئری‌ها:**

* **کاهش زمان پاسخگویی: با کاهش تعداد کوئری‌ها، زمان لازم برای دریافت داده‌ها کاهش می‌یابد.**
* **کاهش بار روی سرور: تعداد کمتر کوئری‌ها به معنای بار کمتر روی سرور پایگاه داده است.**
* **افزایش کارایی: برنامه سریع‌تر و کارآمدتر عمل می‌کند.**

**Lazy Loading: این روش؛ رفتار دیفالت eloquent هستش که؛ داده های مرتبط هر داده اصلی؛ با کوئری های جداگونه دریافت میشه.**

**پس دیگه داده های مرتبط؛ همراه با داده های اصلی دریافت نمیشن و خودمون باید جداگونه دریافتشون کنیم.**

**# نکته ی مهم: Lazy Loading به شما این امکان را می‌دهد که تنها داده‌های مورد نیاز خود را بارگذاری کرده و از بارگذاری داده‌های غیر ضروری جلوگیری کنید.**

**پس؛ layz loading برای گرفتن داده های مرتبط؛ نباید استفاده بشه چون مشکل N+1 Query بوجود میاد.**

**ولی اگه داده های مرتبط رو نمیخوای؛ بهتره که استفاده بشه.**

**این روش؛ برای دریافت داده های مرتبط؛ منجر به مشکلات پرفورمنسی میشه (افزایش تعداد کوئری ها + افزایش زمان پاسخگویی نرم افزار + استفاده ی بیشتر از منابع) که به عنوان N+1**

**query problem شناخته میشه چون به ازای هر داده مرتبط؛ باید یک کوئری جداگونه بزنیم.**

**مثال:**

**$posts = Post::all();**

**foreach ($posts as $post) {**

**$comments = $post->comments; // یک کوئری جداگانه برای هر پست برای بارگذاری نظرات**

**}**

**چه زمانی از Eager Loading و چه زمانی از Lazy Loading استفاده کنیم؟**

**با درک صحیح این دو مفهوم، شما می‌توانید کوئری‌های پایگاه داده را بهینه کرده و عملکرد برنامه خود را بهبود بخشید.**

* **Eager Loading:**
  + **هنگامی که به تمامی داده‌های مرتبط با یک مدل نیاز دارید.**
  + **هنگامی که تعداد روابط بین مدل‌ها کم است.**
  + **هنگامی که عملکرد برنامه برای شما بسیار مهم است و می‌خواهید تعداد کوئری‌های اجرا شده را به حداقل برسانید.**
* **Lazy Loading:**
  + **هنگامی که به تمامی داده‌های مرتبط با یک مدل نیاز ندارید.**
  + **هنگامی که تعداد روابط بین مدل‌ها زیاد است.**
  + **هنگامی که می‌خواهید از بارگذاری داده‌های غیر ضروری جلوگیری کنید.**