选择性加载和预加载属性

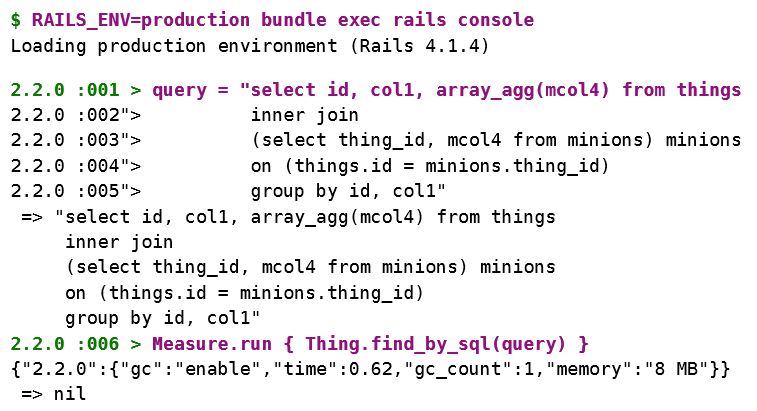
如果从“仅加载您需要的属性”部分获取我的建议，并仅选择我们需要的值，那将会更好。但是有一个问题。Rails没有从依赖模型中选择列子集的便捷方法。例如，这个操作会失败：

Thing.all.includes(:minions).select("col1", "minions.mcol4").load

它失败了，因为includes（：minions）运行了一个额外的查询来获取所选事物的minions。并且Rails不够聪明，无法确定哪些选择值属于Minions表。如果我们从belongs\_to关联的一侧查询，我们需要使用joins。

Minion.where(id: 1).joins(:thing).select("things.col1", "minions.mcol4")

从has\_many侧joins将返回相同Thing对象的重复项，在我们的示例中为10个重复项。为了解决这个问题，我们可以使用PostgreSQL特定的array\_agg功能来聚合连接表中的列数组。



仅仅看一下内存消耗：全部预加载仅需要8MB而不是518MB。最为回报，它运行速度比原来快了20倍。

限制你选择的值的数量可以节省几秒的执行时间和数百兆的内存。

使用Each！Rails模式下为find\_each和find\_in\_batches

实例化很多ActiveRecord模型是很昂贵的。Rails开发人员了解后添加了两个函数来在大型数据集中批量循环。ind\_each和find\_in\_batches都会默认加载1,000个对象并将它们返回给你——第一个函数，一个接一个。后者为一次性加载整个批次。

您可以使用：batch\_size选项请求更小或更大的批次。