2.4 Добављање података

При раду са релационим базама података, поготово у модерним апликацијама, начин на који систем добавља податке игра кључну улогу у његовој перформантности. Два основна механизма за добављање података у модерним објектно-релационим маперима су Lazy Loading и Eager Loading. Иако оба механизма нуде различите приступе у решавању истог проблема, сваки има своје специфичне предности и изазове у коришћењу. Неадекватно коришћење датих механизама може непосредно довести до деградације у перформансама система које могу резултовати непотребним упитима ка бази података, изазивајући додатно опетерећење у систему и продужити време одзива апликације. Стога је од велике важности разумети како ови механизми функционишу, како би правилном применом постигле жељене перформансе система.

2.4.1 Lazy Loading

Lazy Loading је механизам који омогућава учитавање података само када су они заиста потребни. Другим речима, подаци се не довлаче из базе података све док апликација експлицитно не захтева приступ тим подацима, што може значајно побољшати ефикасност система. Међутим, непромишљено коришћење Lazy Loading-а може довести до такозваног *N+1 query* проблема, који ствара значајно оптерећење базе података.

2.4.2 Проблем N+1 упита

Већина модерних ОРМ алата користе Lazy Loading као основно понашање за добављање података. С обзиром на чињеницу да се не добављају подаци које апликација експлицитно не затражи, у тренутку када они постану неопходни за обраду, систем ће слати упите ка бази за сваки неопходан податак. Овакво понашање неретко резултује проблемом N+1 упита ка бази. Ова појава се лако може објаснити на примеру:
Претпоставимо да имамо базу података која опслужује друштвену мрежу у којој су моделоване табеле Објава(Post) и Ознака(Tag). Однос између ових табела је један према више, односно једна објава може да има више ознака.
Уколико би се у систему користио Lazy Loading, добављање података би се свело на:

1. Основни упит којим би се добавиле све објаве
2. За сваку објаву која се добави, прави се додатни упит како би се добавиле све ознаке везане за конкретну објаву.
   За N објава у бази, уместо да се шаље само један упит, ово би резултовало са N+1 упитом ка бази података, одакле и потиче назив проблема. N+1 проблем може да изазове спор одзив система, као и да лимитира скалабилност система, стога је битно узети у обзир неколико могућих солуција. Једна од њих је *Eager Loading*.

2.4.3 Eager Loading

Eager Loading подразумева учитавање свих повезаних података у исто време када и основни објекат. Ово значи да апликација одмах добавља све неопходне релације и податке, минимизујући број упита према бази података. Иако овај приступ решава проблем N+1 упита и користан је када су сви подаци одмах потребни, он такође може представљати оптерећење у случајевима када нису неопходни сви подаци.

Из претходно наведеног може се закључити да избор између механизама Lazy и Eager Loading-а треба да буде базиран на конкретним потребама апликације и перформансама система.