

Propriedades de Navegação em Linq to Objects?

Considere uma aplicação que possui as seguintes classes:

```
class Genero
{
    public int Id { get; set; }
    public string Nome { get; set; }
}

class Musica
{
    public int Id { get; set; }
    public string Nome { get; set; }
    public int GeneroId { get; set; }
}
```

Essa aplicação declara e preenche duas listas dos tipos de classe acima, da seguinte maneira:

```
List<Genero> generos = new List<Genero>
{
    new Genero { Id = 1, Nome = "Rock" },
    new Genero { Id = 2, Nome = "Reggae" },
    new Genero { Id = 3, Nome = "Rock Progressivo" },
    new Genero { Id = 4, Nome = "Jazz" },
    new Genero { Id = 5, Nome = "Punk Rock" },
    new Genero { Id = 6, Nome = "Classica" }
};

List<Musica> musicas = new List<Musica>
{
    new Musica { Id = 1, Nome = "Sweet Child O'Mine", GeneroId = 1 },
    new Musica { Id = 2, Nome = "I Shot The Sheriff", GeneroId = 2 },

    new Musica { Id = 3, Nome = "Danúbio Azul", GeneroId = 6 }
};
```

Suponha que agora você precise criar uma consulta que combine essas duas listas e que traga o id da música, o nome da música e o nome do gênero musical. Então você implementa a consulta conforme abaixo:

```
var query = from m in musicas
join g in generos on m.GeneroId equals g.Id
select new
{
    MusicaId = m.Id,
    Musica = m.Nome,
    Genero = g.Nome
};
```

Uma colega vê o seu código e pergunta se você não poderia reescrever sua consulta, substituindo o join por uma propriedade de navegação. O que você responderia? Justifique sua resposta.