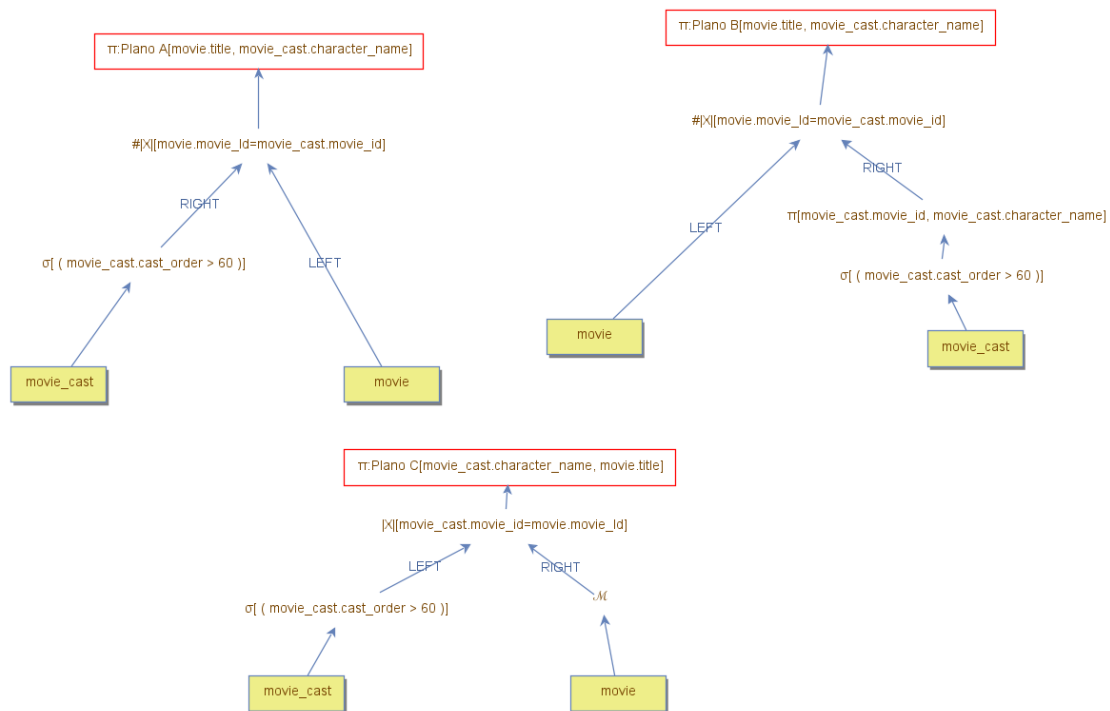


## Aluno: Diego Rockenbach



| Total tuples loaded: 33 |         |         |         |
|-------------------------|---------|---------|---------|
|                         | Plano C | Plano B | Plano A |
| Tuples loaded           | 11      | 11      | 11      |
| Accessed blocks         | 544     | 541     | 542     |
| Loaded blocks           | 0       | 0       | 0       |
| Saved blocks            | 0       | 0       | 0       |
| Filter comparisons      | 3179    | 3179    | 3179    |
| Memory Used             | 637     | 737     | 1738    |
| Next Calls              | 73      | 295     | 284     |
| Primary key searches    | 7       | 0       | 0       |
| Records Read            | 3193    | 3419    | 3419    |
| Sorted tuples           | 0       | 0       | 0       |

Como podemos observar, o plano B possui o menor número de blocos acessados, seguido do plano A e então do C. Em questão de memória usada, o plano A é o mais custoso, seguido do B e então do C.

Analisando esses resultados, e tendo como principal medidor de performance a quantidade de blocos acessados, vemos que fazer uma projeção intermediária, logo

depois do *filter* de *movie\_cast* foi mais eficiente, pois reduz a quantidade de atributos carregados na memória durante o filtro e junção.

O plano A primeiro aplica o filtro em *movie\_cast*, e depois junta *movie* e *movie\_cast* com um *hash join*, para só então, no final do plano, realizar a projeção. Apesar do desempenho ter sido muito próximo do plano B, tendo acessado somente 1 bloco a mais, seu aproveitamento de memória foi muito inferior, pois muitos dados intermediários foram carregados antes da projeção ser aplicada.

O plano C, por sua vez, utiliza *memoise* na tabela *movie*, e utiliza um *nested loop* ao invés de um *hash join*, como era o caso dos outros dois planos, para então, novamente, ter sua única projeção no final do plano. Apesar do uso de memória ter sido menor que os outros dois planos, ele também possuiu o pior desempenho em relação a blocos acessados, tendo acessado 3 blocos a mais que o plano B e 2 blocos a mais que o plano A.

Assim, a decisão de qual escolher irá depender de qual finalidade buscamos. Em um sistema com pouca memória disponível, a perda de performance em questão de blocos acessados do plano C pode se pagar se comparado ao seu baixo uso de memória. Em contraponto, caso memória não seja um problema, e deseje-se apenas acessar o menor número possível de blocos, o plano B será a escolha superior.