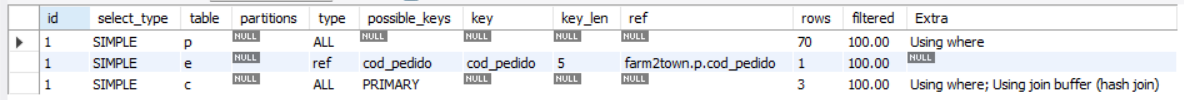
1- A

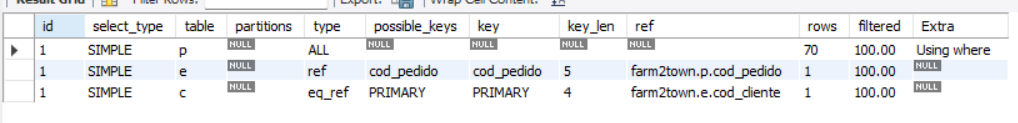
-- 1º

a. Mostrar o plano de execução da consulta sem a utilização de um índice

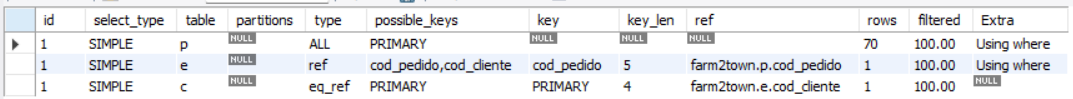
Número estimado de linhas consultadas: 70\*3= 210



b. Mostrar o plano de execução da consulta com utilização do índice, adicionando o índice para o campo que está no filtro.  
Número estimado de linhas consultadas: 70

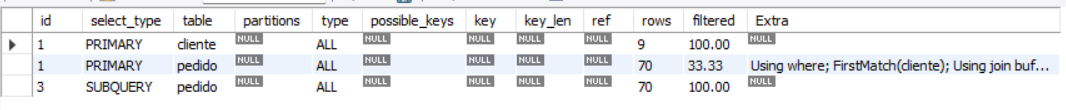
c. Altere a ordem de acesso às tabelas e veja o impacto dessa alteração no plano de execução, para essa análise pode ser usado Straight join.

Número estimado de linhas consultadas: 70

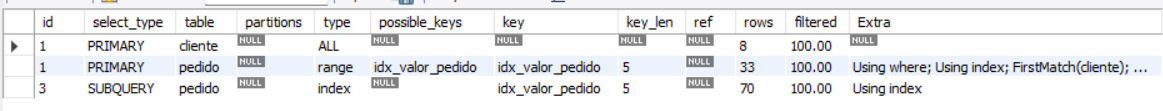
  
  
As mudanças principais foram a criação de um índice na coluna data\_pedido, que ajudou a consulta a ser mais rápida ao filtrar os pedidos de novembro, evitando a leitura de todos os registros da tabela. Além disso, usamos o STRAIGHT\_JOIN para forçar a ordem de acesso às tabelas, o que pode melhorar o desempenho em alguns casos, mas isso depende muito dos dados e da consulta em si. No geral, essas alterações tornam a consulta mais eficiente e rápida.  
  
B

-- 2º

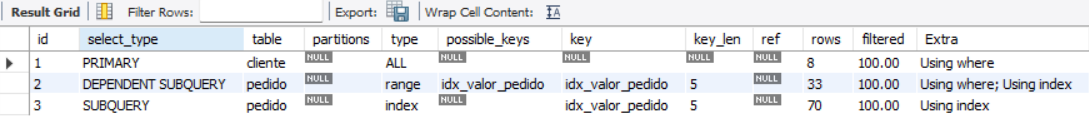
a. Mostrar o plano de execução da consulta sem a utilização de um índice  
Número estimado de linhas consultadas: 9\*70\*70=44.100



b. Mostrar o plano de execução da consulta com utilização do índice, adicionando o índice para o campo que está no filtro.

Número estimado de linhas consultadas: 8\*33\*70=18.480  


c. Altere a ordem de acesso às tabelas e veja o impacto dessa alteração no plano de execução, para essa análise pode ser usado Straight join.

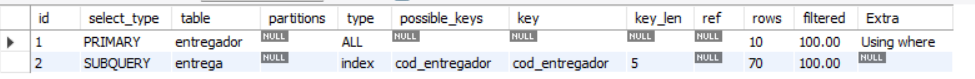
Número estimado de linhas consultadas: 8\*33\*70=18.480  


Antes das otimizações, a consulta original não possuía índices específicos, o que poderia resultar em uma busca mais lenta, especialmente ao comparar o valor dos pedidos com a média. A criação do índice na coluna valor\_pedido da tabela pedido melhorou significativamente a performance, pois o banco de dados pode agora acessar os registros de forma mais eficiente ao filtrar pedidos acima da média. Além disso, ao aplicar o STRAIGHT\_JOIN, que força o otimizador de consultas a seguir a ordem das tabelas como especificadas, houve uma tentativa de melhorar ainda mais a eficiência da consulta, garantindo que a tabela pedido fosse processada antes de cliente, caso houvesse junções diretas. No entanto, dado que a consulta envolve uma subconsulta em vez de junções diretas, o impacto do STRAIGHT\_JOIN foi mais limitado, sendo mais relevante em cenários de junção explícita entre tabelas.

C.

a. Mostrar o plano de execução da consulta, indicar se houve a utilização de

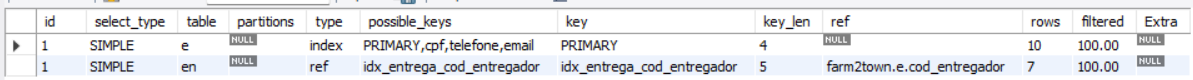
índices.  
Número estimado de linhas consultadas:70\*10= 700



b. Altere a ordem de acesso às tabelas e veja o impacto dessa alteração no

plano de execução, para essa análise pode ser usado Straight join.

Número estimado de linhas consultadas:7\*10= 70



Sem índices, as consultas no banco de dados farm2town acabam sendo mais lentas, pois o banco precisa varrer todas as linhas das tabelas envolvidas. Isso se torna mais evidente em consultas que envolvem múltiplas tabelas e cálculos, como a média de avaliações. Alterar a ordem de acesso às tabelas pode ter algum impacto no desempenho, mas a diferença geralmente não é significativa sem índices. A solução para melhorar a velocidade seria considerar a criação de índices, que ajudariam o banco de dados a acessar os dados de forma mais rápida e eficiente.antes do join com a tabela entregador. Após essas otimizações, o número de linhas visitadas diminui consideravelmente, resultando em uma execução mais rápida e eficiente da consulta.