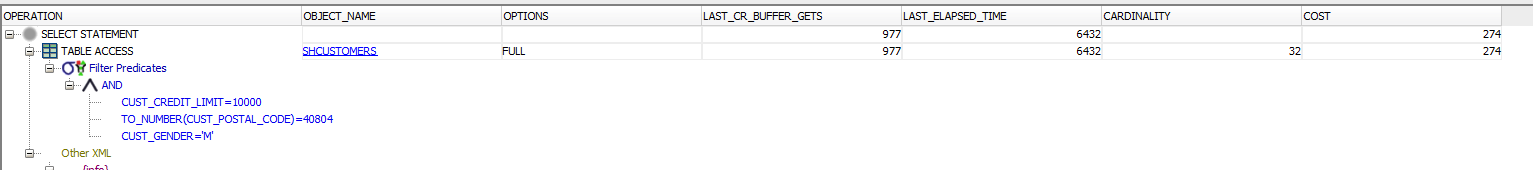
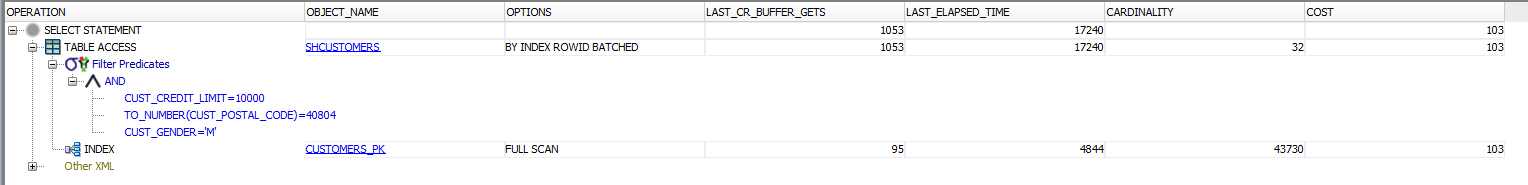
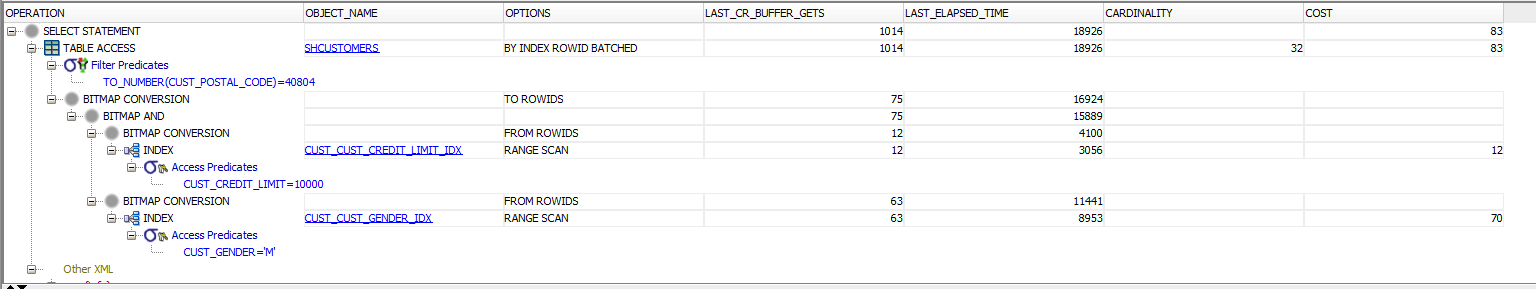
**2. B-деревья**

**B)** 

Запрос выполняет полное сканирование таблицы для получения данных

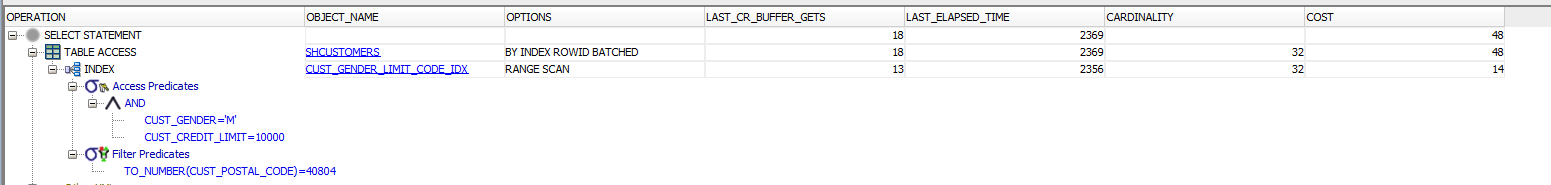
**C)** 

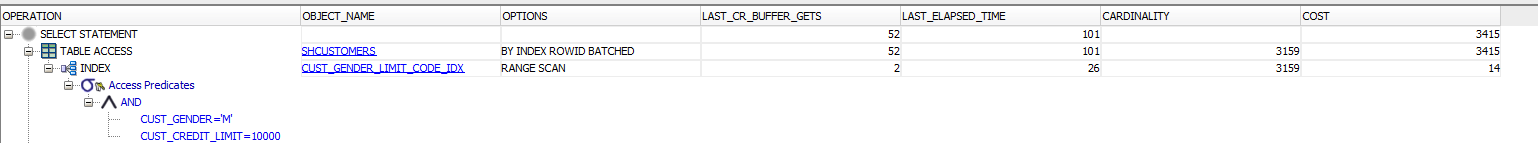
При получении информации использовался индекс, что позволило снизить стоимость запроса более чем в 2 раза



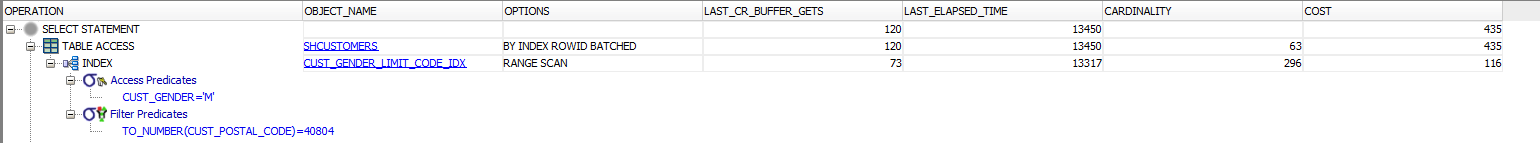
При комбинировании индексов стоимость упала почти в 4 раза по сравнению с полным сканированием

**Индекс по трем полям**

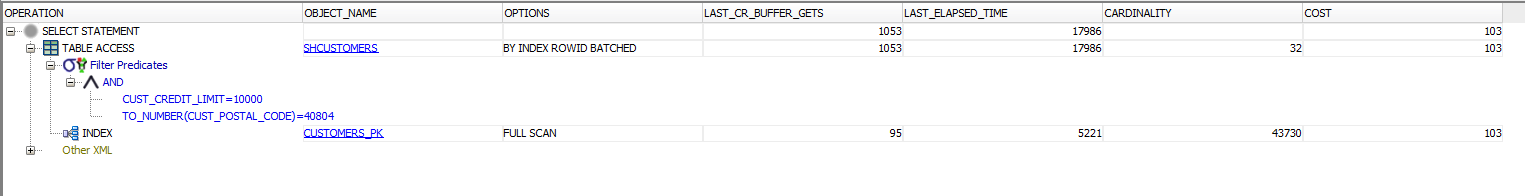




Стоимость и кардинальность запроса выросли почти на два порядка при отсутствии третьего поля индекса



Стоимость и кардинальность запроса все еще выше, чем у запроса с тремя полями из индекса

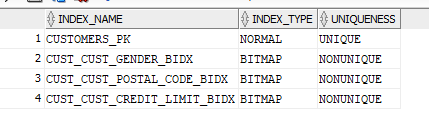


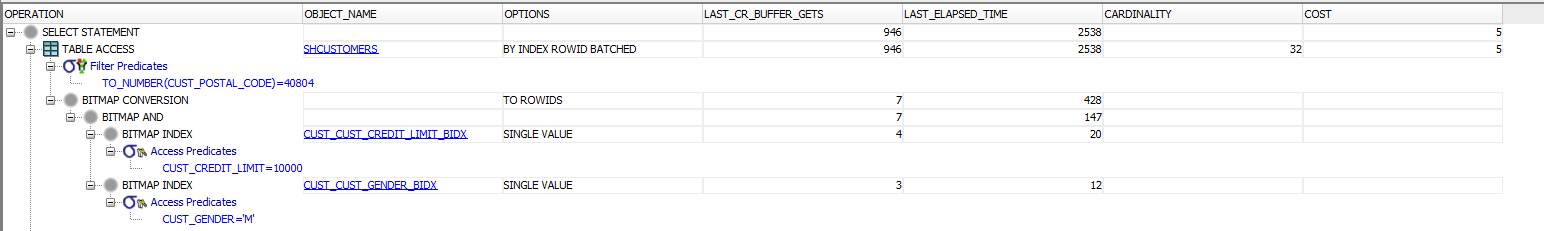
Получили кардинальность равную запросу с тремя полями при стоимости в два раза выше

Расхождения объясняются структурой индекса. При выборке по трем полям, индекс позволил быстро найти данные, в то время как в других запросах необходимо было просматривать больше записей.

Расхождения в трех измененных запросах объясняются тем, что поля имеют разный набор возможных значений, а потому меняется сложность поиска информации в дереве.

**3. BitMap индексы**





В тексте лабораторной не очень ясно, что необходимо сделать с битмап индексами (в пункте просто даны запросы для их создания). Результат прогона

SELECT /\*+ INDEX\_COMBINE(c) \*/ c.\*

FROM Shcustomers c

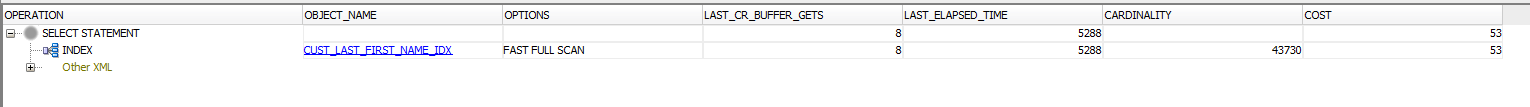
WHERE cust\_gender = 'M'

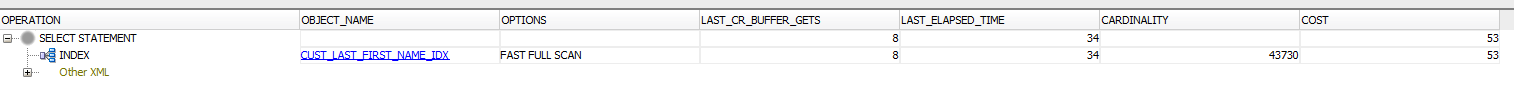
AND cust\_postal\_code = 40804

AND cust\_credit\_limit = 10000;

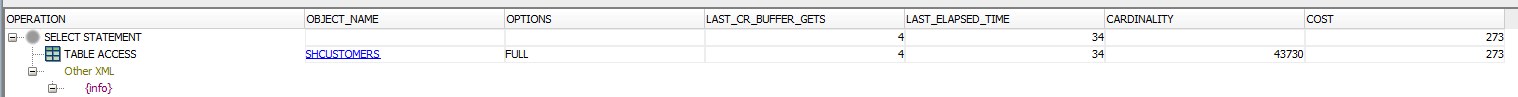
приведен выше. По сравнению с тем же запросом с обычными индексами, скорость получения данных сильно возросла.

**4. Использование индексов для выборки данных**

**G)** 

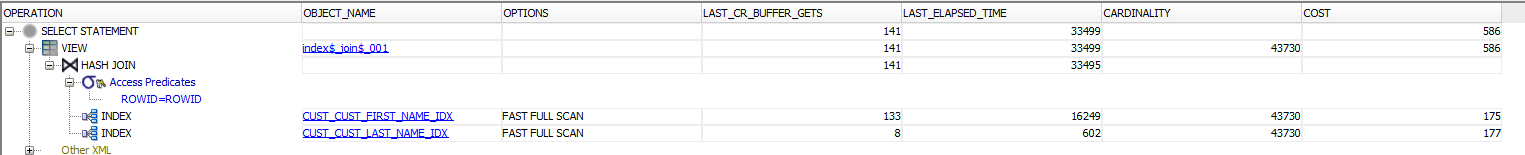


При изменении порядка следования полей разницы практически нет

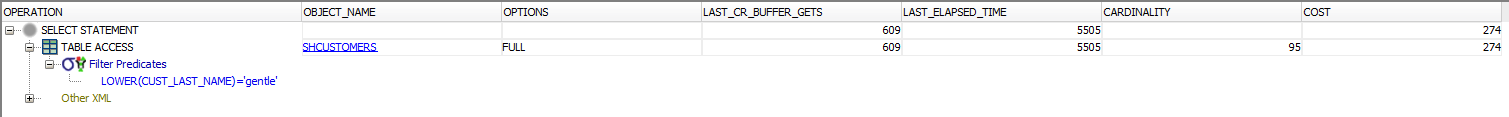


А при добавлении поля в запрос выборка данных происходит с полным сканированием, а не с использованием индекса

**H)**

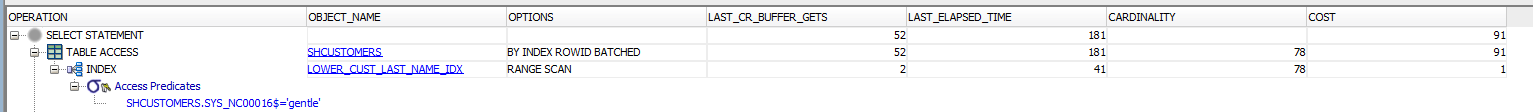


Кардинальность запроса осталась той же, но стоимость и время значительно возросли из-за объединения индексов

**I)** 

Индекс не используется, так как в запросе поле обернуто в функцию LOWER

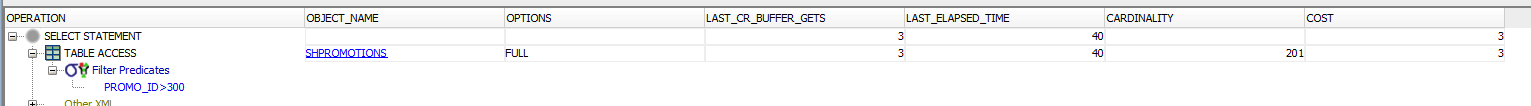
**J)**

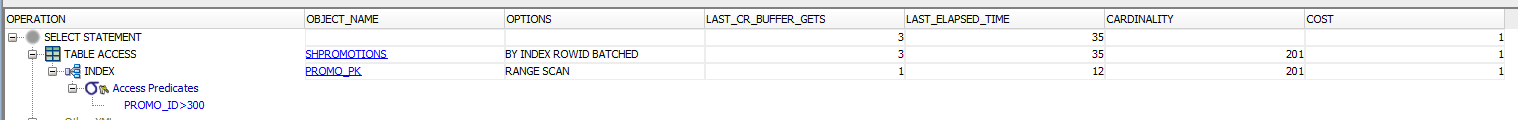


После создания индекса по функции, в данном запросе он подключается, что приводит к существенному ускорению выборки и уменьшению ее стоимости.

**5. Сравнение индексов и таблиц, организованных по индексу.**

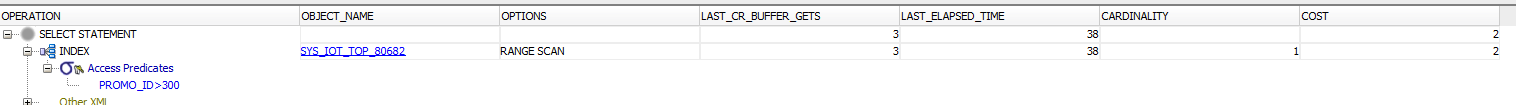
**K)**





При подключении индекса запрос получился дешевле и быстрее, чем без него.

**L)**

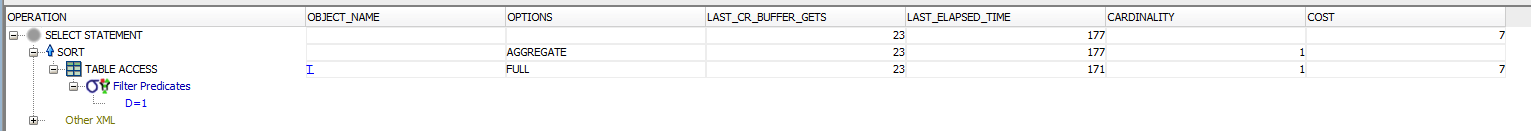


По производительности запрос оказался между запросом на обычной таблице без и с индексом.

План с индексом на обычной таблице оказался лучше

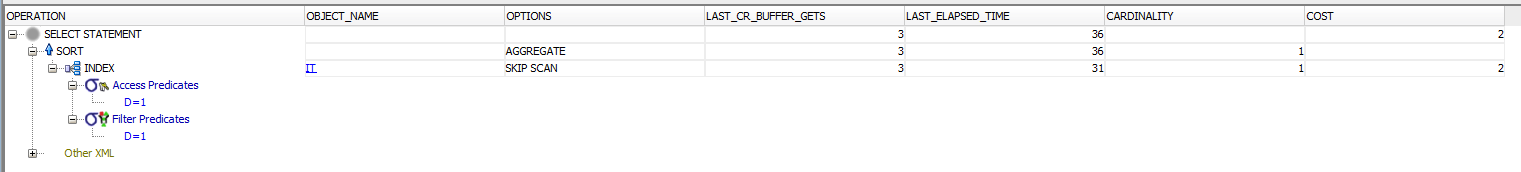
**6. Сканирование с пропуском**

**M)**



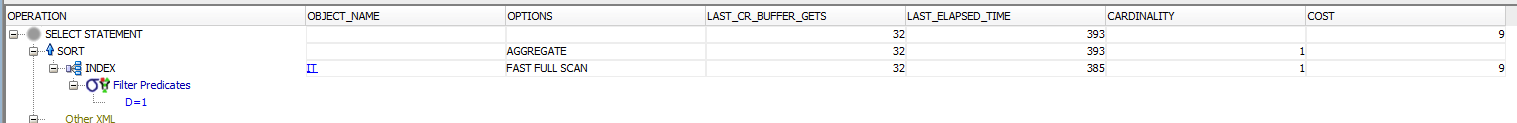
Индекс не подключается, так как оптимизатор не знает, сколько записей возможно найти

**N)**



После сбора статистики оптимизатор принял решение использовать пропускающее сканирование

O)



При использовании индекса без пропусков, а с быстрым полным сканированием, стоимость и кардинальность возросли из-за сканирования целиком.