* Cambios en la base de datos:

Agregamos el atributo ID\_COMPAÑIA a la tabla FUNCIONES

* A continuación, se presenta la consulta para el requerimiento RFC9 para la compañía ‘Los teatreros’:

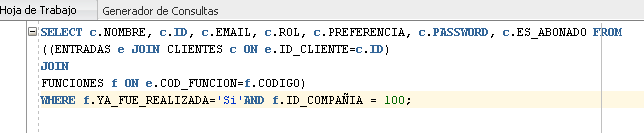
SELECT c.NOMBRE, c.ID, c.EMAIL, c.ROL, c.PREFERENCIA, c.PASSWORD, c.ES\_ABONADO FROM

((ENTRADAS e JOIN CLIENTES c ON e.ID\_CLIENTE=c.ID)

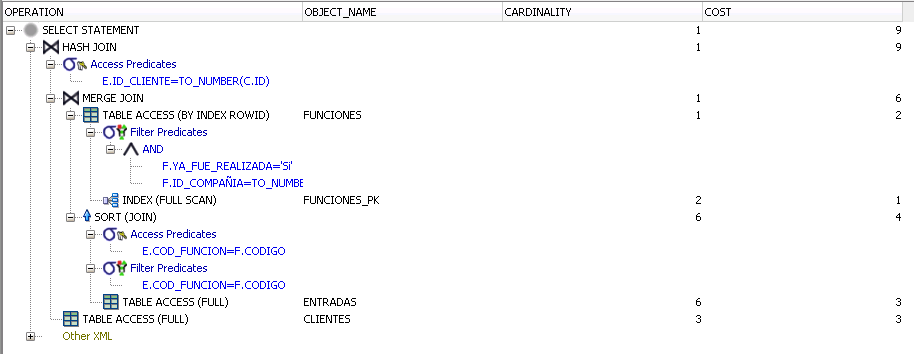
JOIN

FUNCIONES f ON e.COD\_FUNCION=f.CODIGO)

WHERE f.YA\_FUE\_REALIZADA='Si'AND f.ID\_COMPAÑIA = 100;



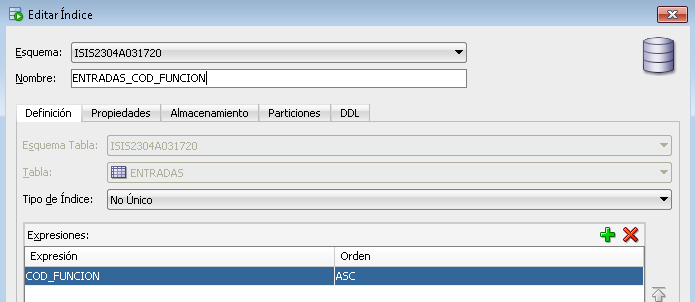
Plan de ejecución con índices generados automáticamente por Oracle

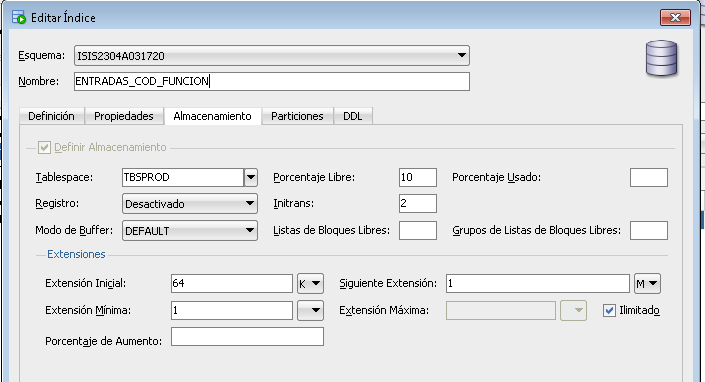


Basándonos en esta consulta, vemos que resulta factible crear índices sobre los siguientes atributos:

* COD\_FUNCION de la tabla ENTRADAS



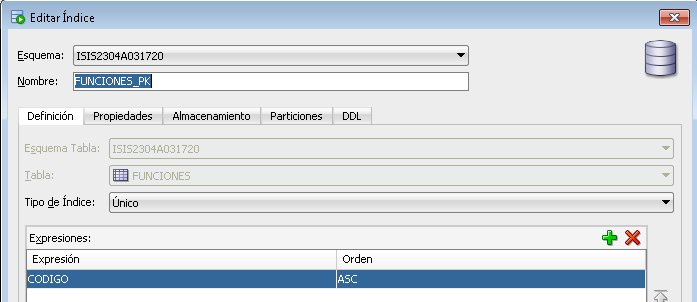


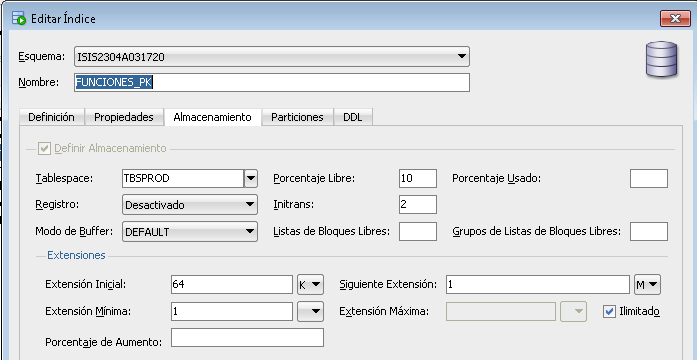


Justificación: El segundo Join se hace en base a la condición e.COD\_FUNCION=f.CODIGO por lo tanto, al implementar índices el requerimiento se optimiza y el costo del plan de ejecución disminuye, como vimos en clase, es recomendable usar índices sobre los atributos que determinan la condición del join.



* YA\_FUE\_REALIZADA de la tabla FUNCIONES  
  



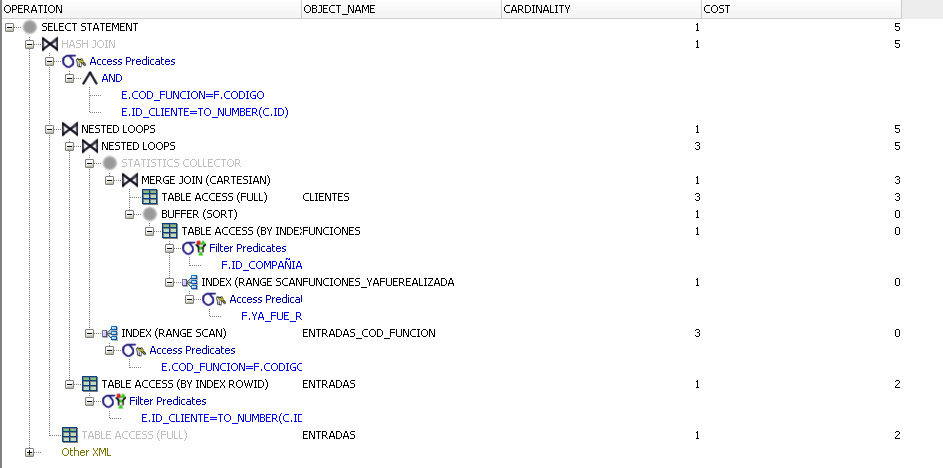


Justificación:

Las condiciones del WHERE contienen el atributo YA\_FUE\_REALIZADA por lo tanto al

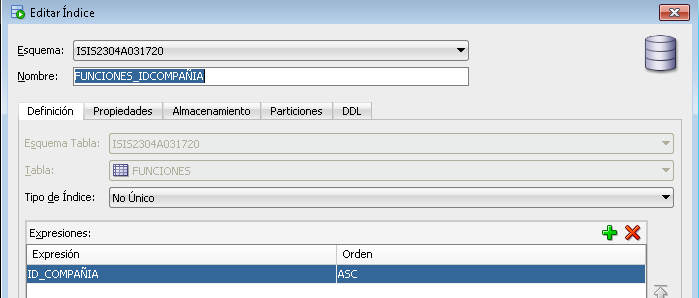
Incluir índice, el rendimiento del algoritmo mejora y el costo del plan de ejecución

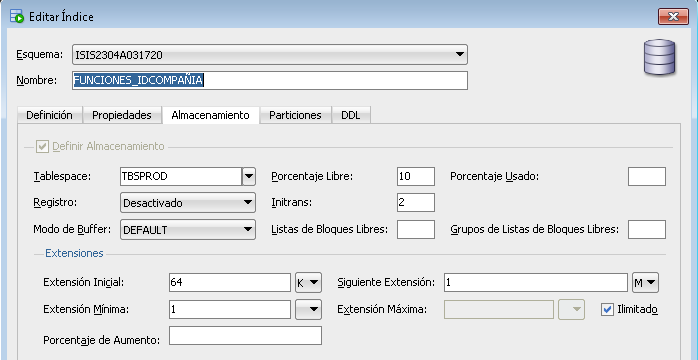
disminuye.



* ID\_COMPAÑIA de la tabla FUNCIONES





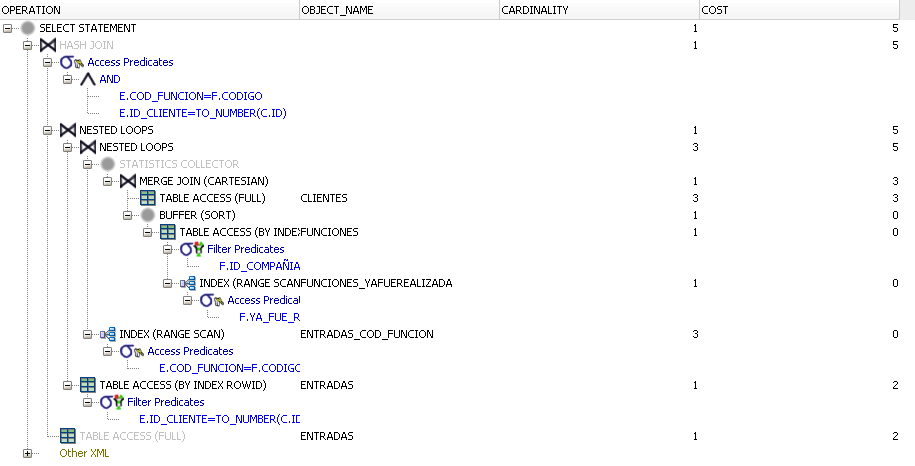


Justificación:

Las condiciones del WHERE contienen el atributo ID\_COMPAÑIA por lo tanto al

Incluir índice, el rendimiento del algoritmo mejora y el costo del plan de ejecución

disminuye.



* Para el requerimiento RFC10 aplican los mismos índices y las mismas justificaciones ya que básicamente es la misma consulta solo que retorna los clientes que no asistieron a dicha función.
* A continuación, se presenta la consulta para el requerimiento RFC11:

SELECT w.FECHA, w.NUM\_ENTRADAS\_VENDIDAS, w.NOMBRE\_ESPECTACULO, sc.NOMBRE AS NOMBRE\_ESCENARIO FROM

(SELECT x.COD\_FUNCION, x.FECHA, x.NUM\_ENTRADAS\_VENDIDAS, es.NOMBRE AS NOMBRE\_ESPECTACULO, x.CODIGO\_ESCENARIO FROM

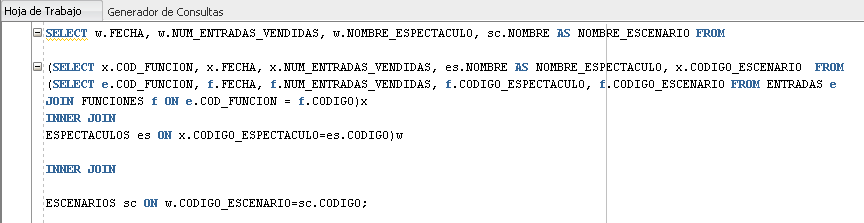
(SELECT e.COD\_FUNCION, f.FECHA, f.NUM\_ENTRADAS\_VENDIDAS, f.CODIGO\_ESPECTACULO, f.CODIGO\_ESCENARIO FROM ENTRADAS e JOIN FUNCIONES f ON e.COD\_FUNCION = f.CODIGO)x

INNER JOIN

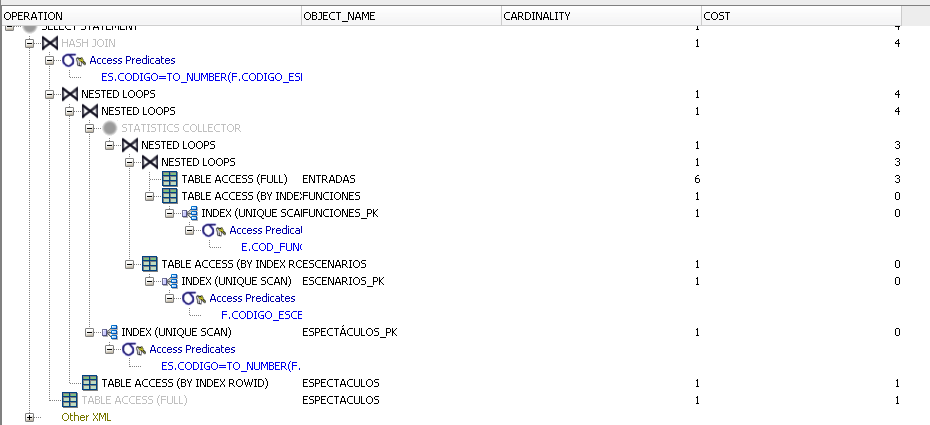
ESPECTACULOS es ON x.CODIGO\_ESPECTACULO=es.CODIGO)w

INNER JOIN

ESCENARIOS sc ON w.CODIGO\_ESCENARIO=sc.CODIGO;



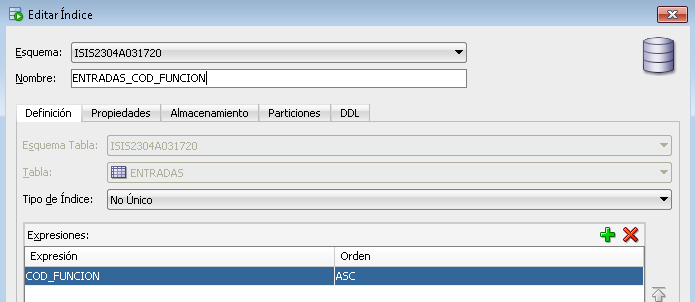
Plan de ejecución con índices generados automáticamente por Oracle:

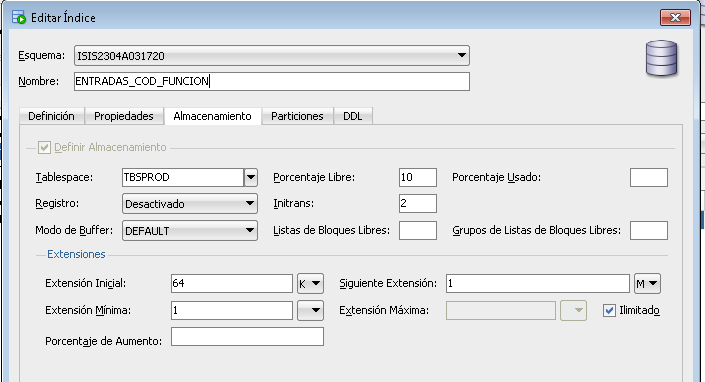


Basándonos en esta consulta, vemos que resulta factible crear índices sobre los siguientes atributos:

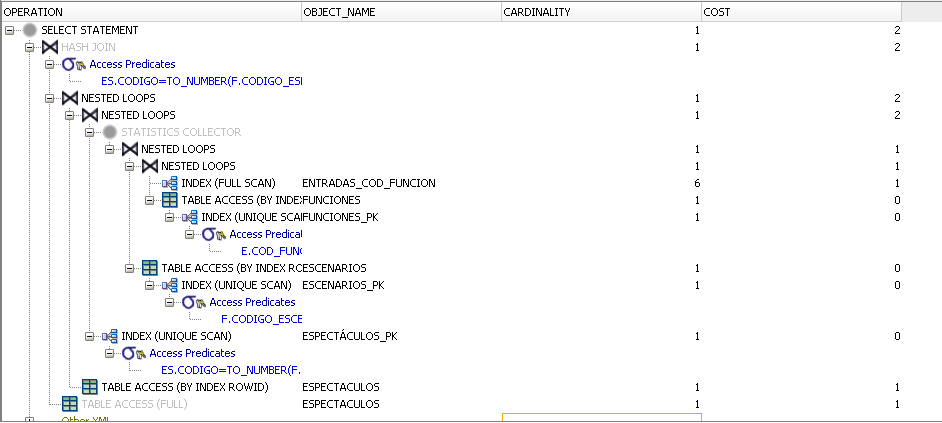
* COD\_FUNCION de la tabla ENTRADAS





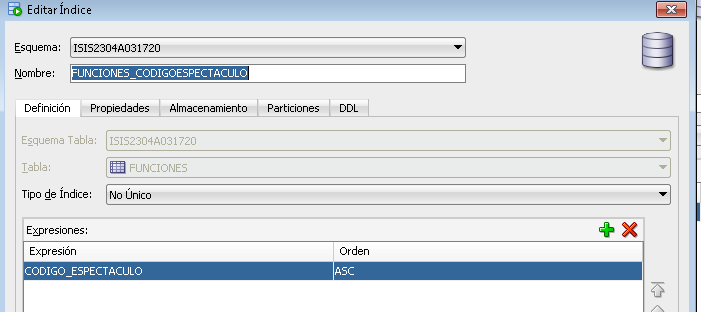


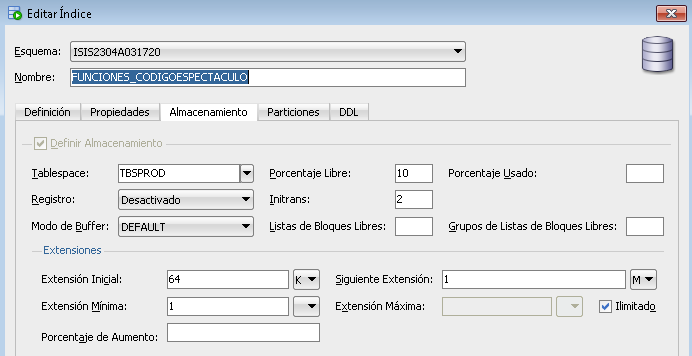
Justificación: El primer Join se hace en base a la condición e.COD\_FUNCION=f.CODIGO por lo tanto, al implementar índices el requerimiento se optimiza y el costo del plan de ejecución disminuye, como vimos en clase, es recomendable usar índices sobre los atributos que determinan la condición del join.



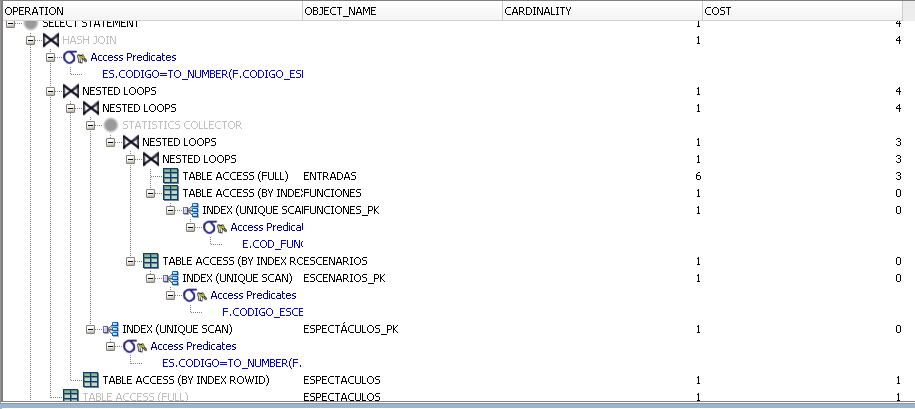
* CODIGO\_ESPECTACULO de la tabla FUNCIONES





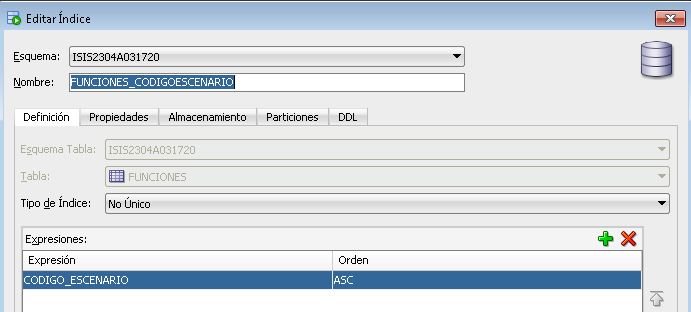


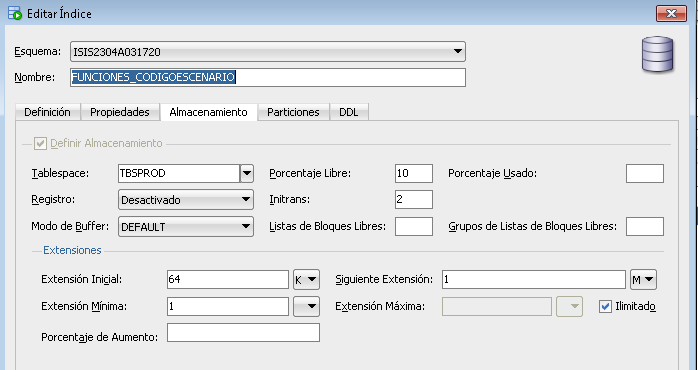
Justificación: El segundo Join se hace en base a la condición x.CODIGO\_ESPECTACULO=es.CODIGO por lo tanto, al implementar índices el requerimiento se optimiza y el costo del plan de ejecución disminuye, como vimos en clase, es recomendable usar índices sobre los atributos que determinan la condición del join.



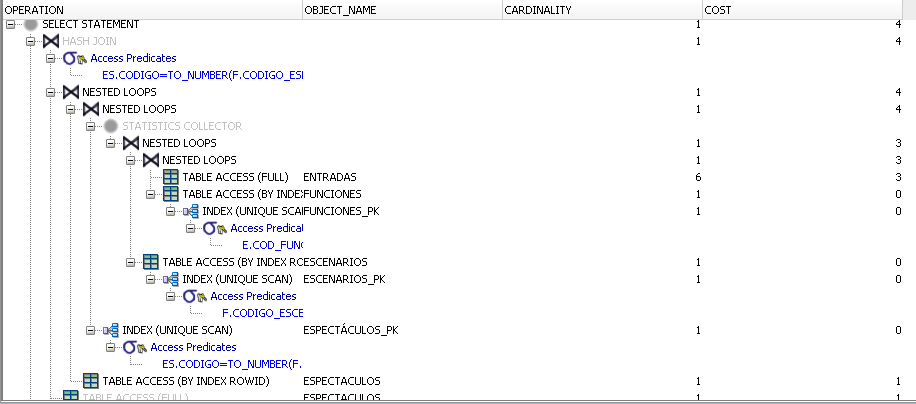
* CODIGO\_ESCENARIO de la tabla FUNCIONES







Justificación: El tercer Join se hace en base a la condición w.CODIGO\_ESCENARIO=sc.CODIGO por lo tanto, al implementar índices el requerimiento se optimiza y el costo del plan de ejecución disminuye, como vimos en clase, es recomendable usar índices sobre los atributos que determinan la condición del join.



A parte el requerimiento dice que Esta consulta puede ser filtrada por diferentes conceptos (rangos de fecha, elementos del escenario, tipo de localidad, franja horaria y día de la semana) de modo que según sea el filtro se pueden crear índices para agilizar la consulta. Por lo tanto si se desea filtrar por rangos de fecha aplicamos índices sobre la columna FECHA de la tabla FUNCIONES y de este modo optimizamos la consulta, lo mismo para el resto de filtros.

* A continuación, se presenta la sentencia que realiza la consulta para el requerimiento RFC12 con n=3:

SELECT b.ID\_CLIENTE FROM

(SELECT en.ID\_CLIENTE, s.LOCALIDAD FROM

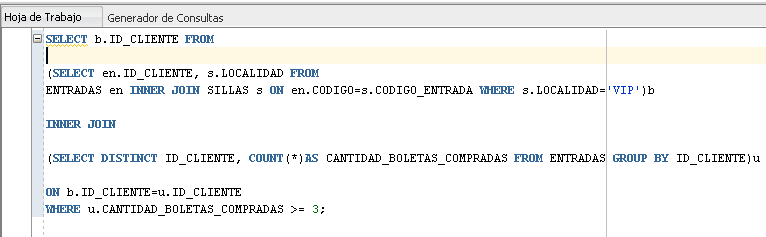
ENTRADAS en INNER JOIN SILLAS s ON en.CODIGO=s.CODIGO\_ENTRADA WHERE s.LOCALIDAD='VIP')b

INNER JOIN

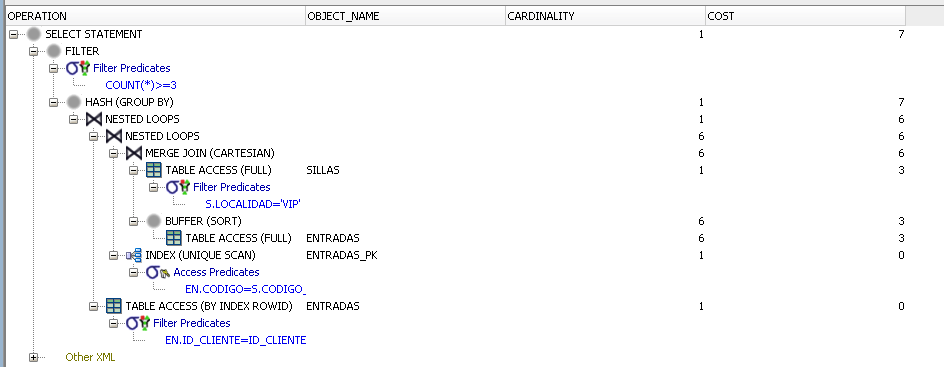
(SELECT DISTINCT ID\_CLIENTE, COUNT(\*)AS CANTIDAD\_BOLETAS\_COMPRADAS FROM ENTRADAS GROUP BY ID\_CLIENTE)u

ON b.ID\_CLIENTE=u.ID\_CLIENTE

WHERE u.CANTIDAD\_BOLETAS\_COMPRADAS >= 3;



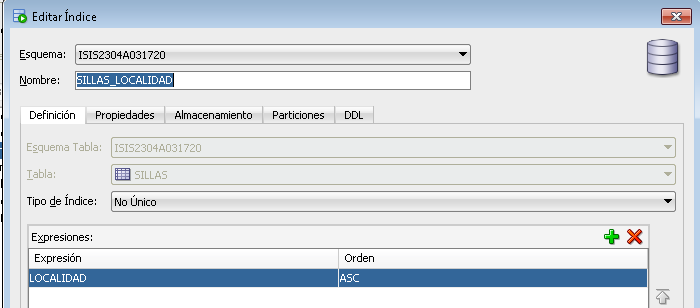
Plan de ejecución con índices generados automáticamente por Oracle:

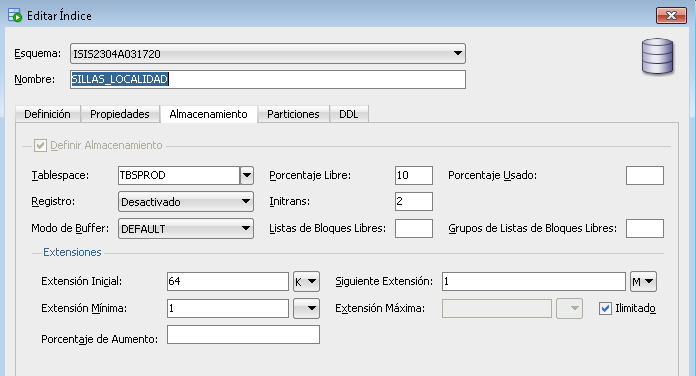


Basándonos en esta consulta, vemos que resulta factible crear índices sobre los siguientes atributos:

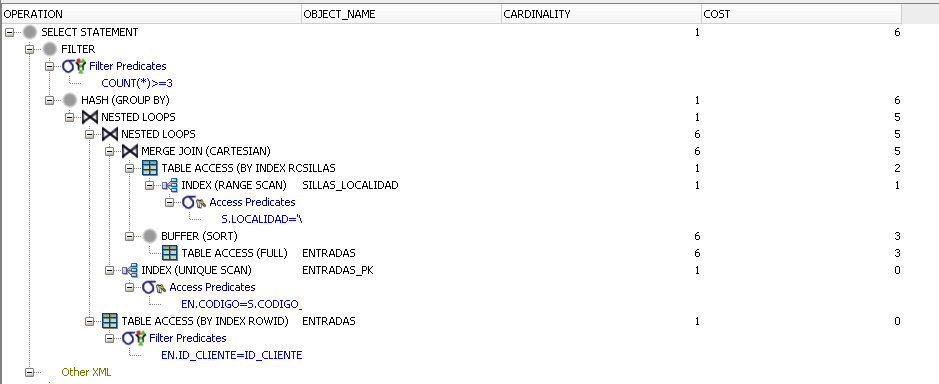
* LOCALIDAD de la tabla SILLAS





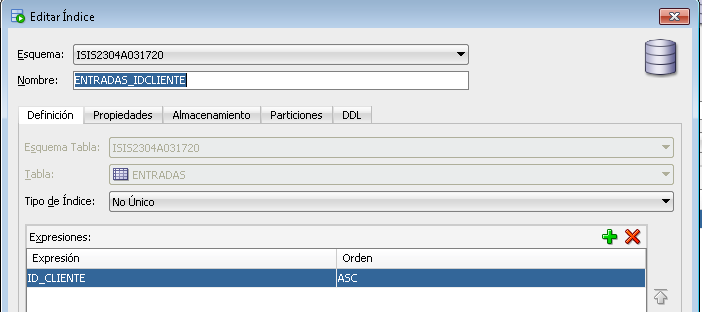


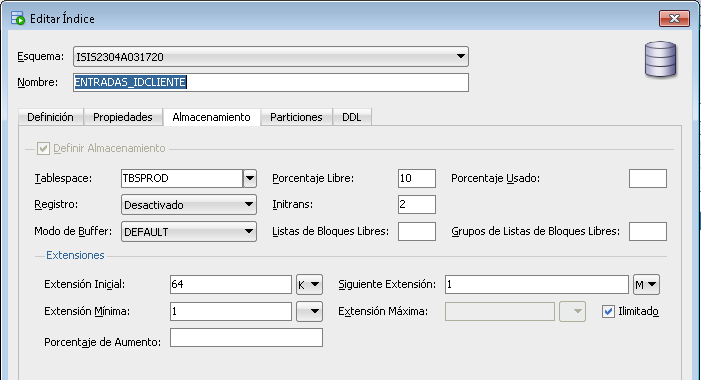
Justificación: En este caso creamos un índice sobre la columna LOCALIDAD de la tabla SILLAS porque en la consulta se filtra la búsqueda sobre aquellas tuplas cuya localidad sea ‘VIP’. Por consiguiente al crear un índice sobre este atributo la consulta se optimizará y el costo del plan de ejecución disminuirá



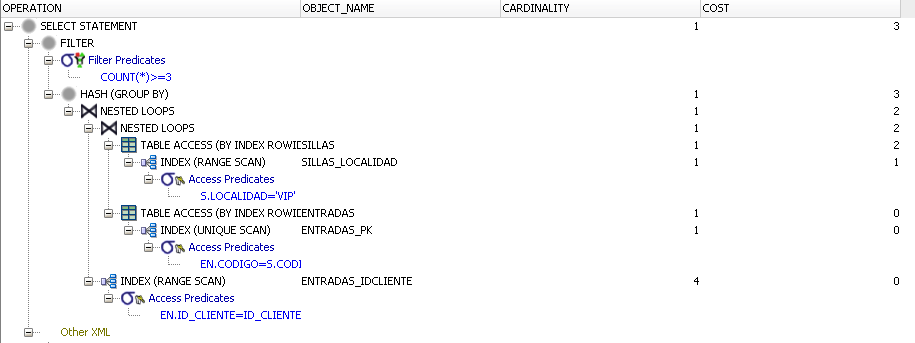
* ID\_CLIENTE de la tabla ENTRADAS







Justificación: En este caso creamos un índice sobre la columna ID\_CLIENTE de la tabla ENTRADAS porque en la consulta se hace un Join en base a la igualdad entre este atributo y el id de la tabla de clientes. Por consiguiente, al crear un índice sobre este atributo la consulta se optimizará y el costo del plan de ejecución disminuirá y que, como vimos en clase, resulta factible usar índices sobre las columnas que están en la condición de un join.

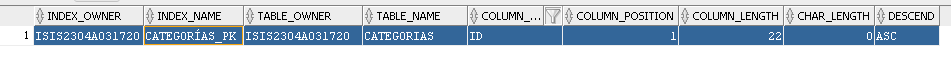


Para todos los casos anteriores se aplicaron índices secundarios, ya que las tablas no caben en memoria principal se necesitan índices que no contengan los registros físicamente sino que sean apuntadores al disco y a demás son índices basados en Hashing como se puede observar en los pantallazos de los planes de ejecución.

* ÍNDICES CREADOS AUTOMÁTICAMENTE POR ORACLE:

Oracle crea automáticamente índices sobre todas aquellas columnas que son llave primaria de las tablas





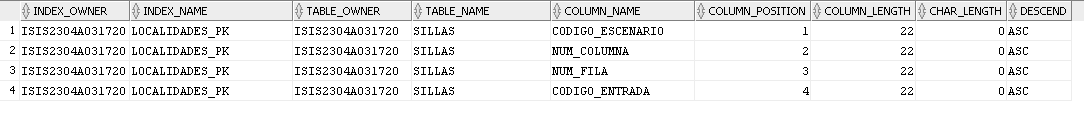












Análisis:

Estos índices fueron creados automáticamente por Oracle porque estas columnas son las llaves primarias de su tabla, por lo tanto, la creación de índices sobre estas columnas tiene mucho sentido ya que la llave primaria de una tabla es el atributo por el cual la gran mayoría de las consultas se guían. Podemos ver claramente que el rendimiento de las consultas mejora notablemente con la implementación automática de estos índices porque están sobre el atributo que identifica como única a cada tupla de cada tabla.