Trabajo 4

Índices Bitmap vs Índices B

Facultad De Ingeniería, Universidad De Cuenca

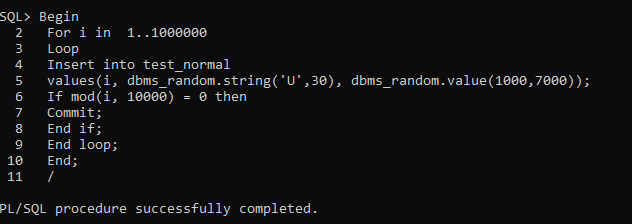
BASES DE DATOS 2

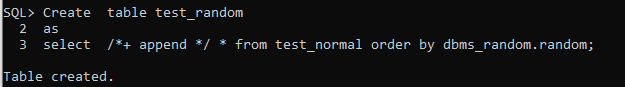
Freddy L. Abad L.

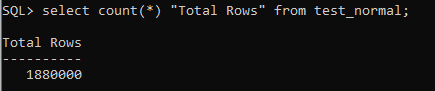
[{freddy.abadl}@ucuenca.edu.ec](mailto:freddy.abadl@ucuenca.edu.ec)

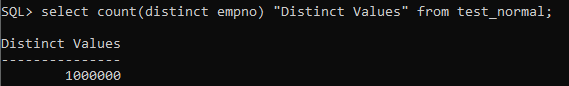
──────────────── ♦️ ────────────────

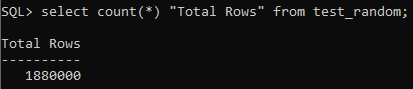


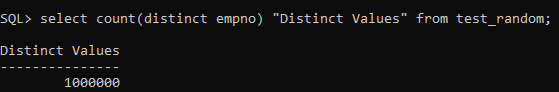










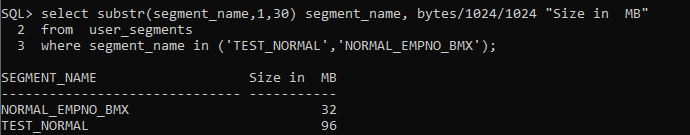


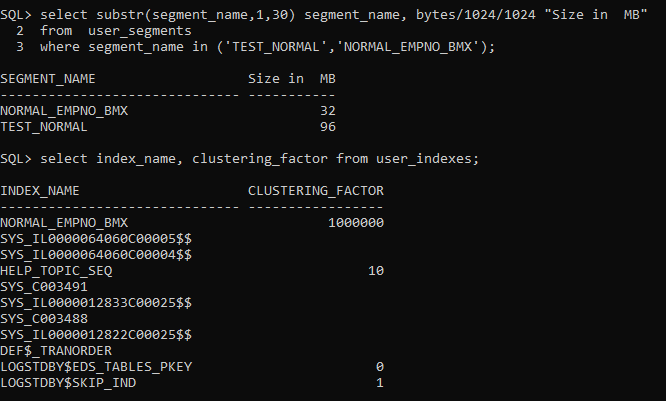
**Paso 1A (con TEST\_NORMAL)**

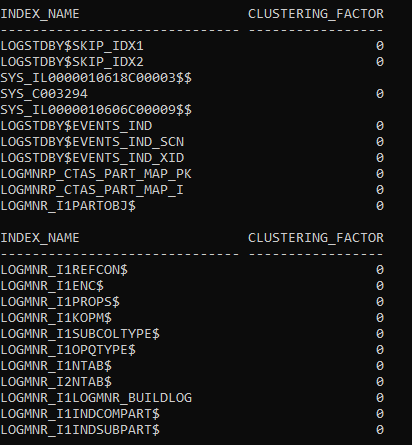
En este paso, crearemos un índice de mapa de bits para la tabla TEST\_NORMAL y, a continuación, verificaremos el tamaño del índice, su factor de agrupamiento [*clustering factor*] y el tamaño de la tabla. Luego ejecutaremos algunas consultas con predicados de igualdad y registraremos las E/S de las consultas empleando el índice de mapa de bits.

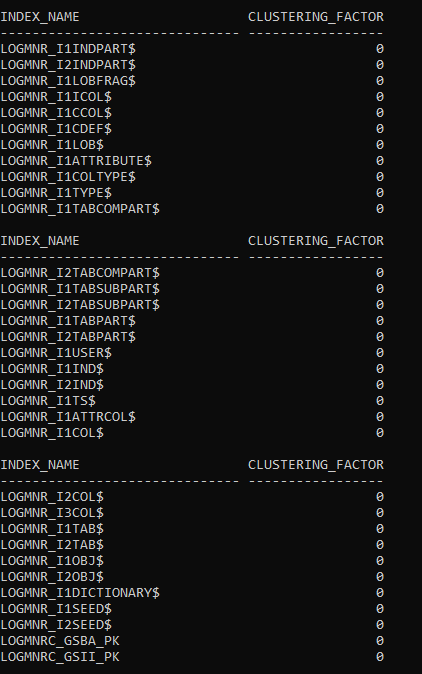
****

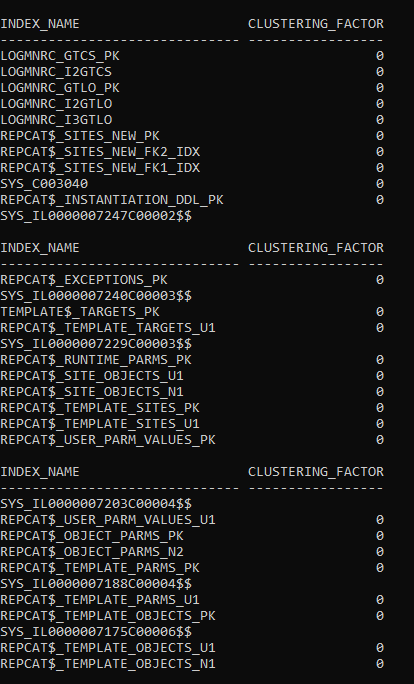
****

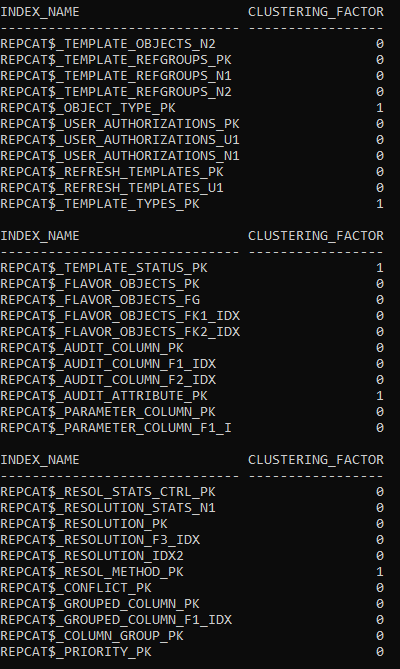
****

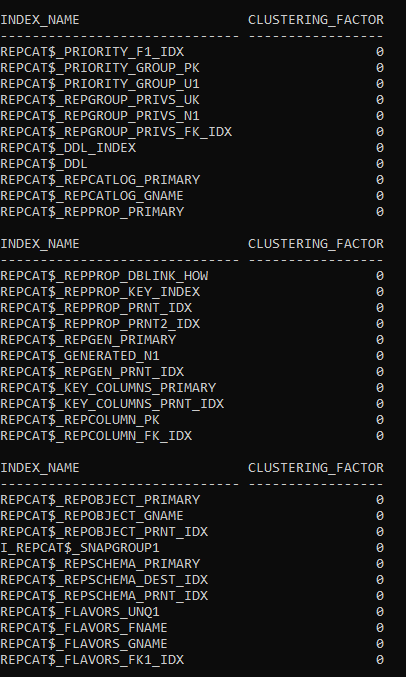
****

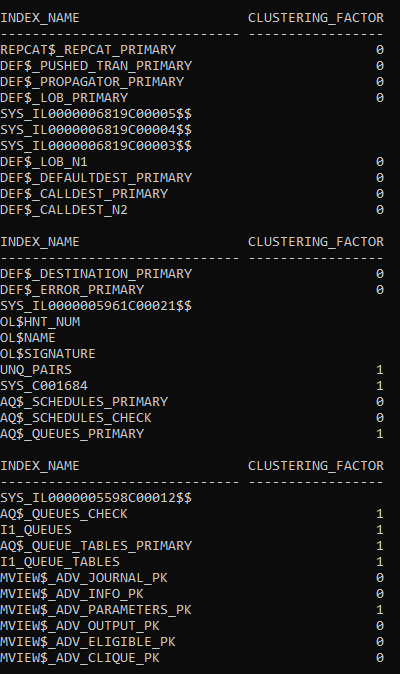
****

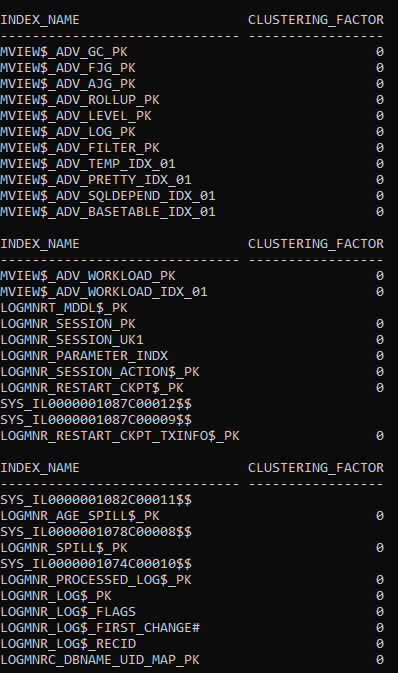
****

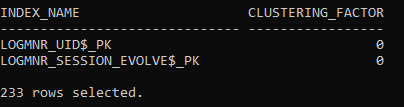
****

****

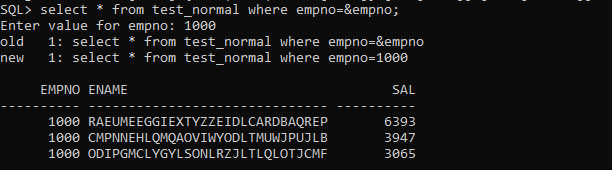
****

****

****

****

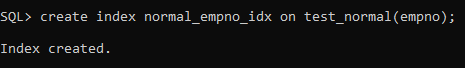
****

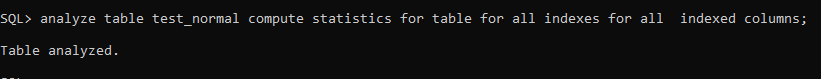
****

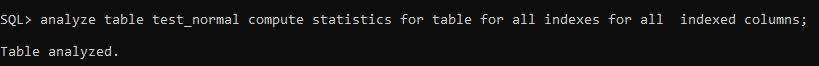
**Paso 1B (con TEST\_NORMAL)**

Ahora quitaremos el índice de mapa de bits y crearemos un índice de árbol B asociado a la columna EMPNO. Como antes, verificaremos el tamaño del índice y el factor de agrupamiento, y ejecutaremos algunas consultas para el mismo conjunto de valores a fin de comparar las E/S.

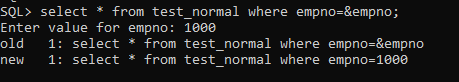
****

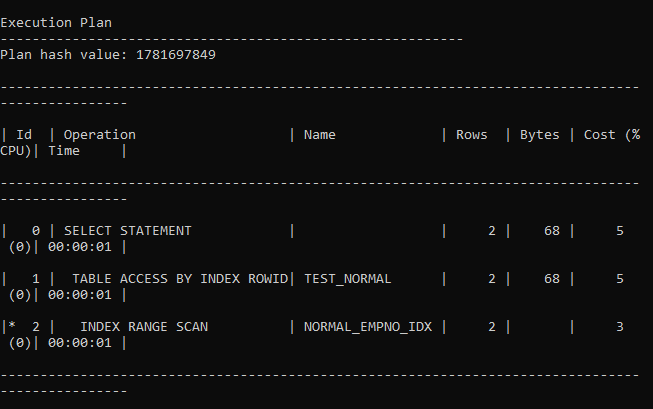
****

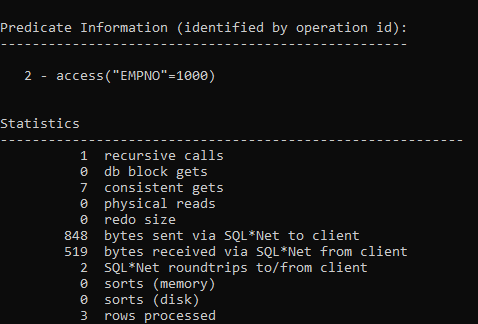
****

****

****

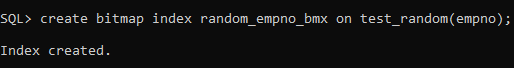
****

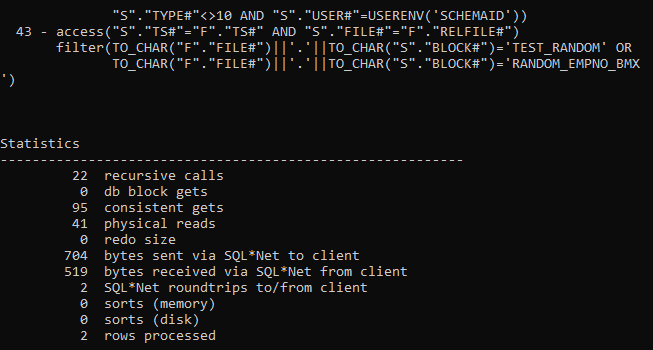
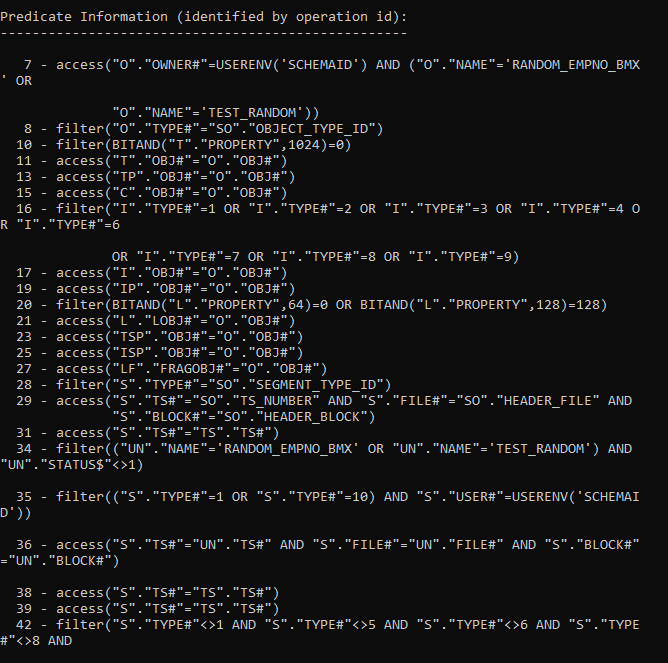
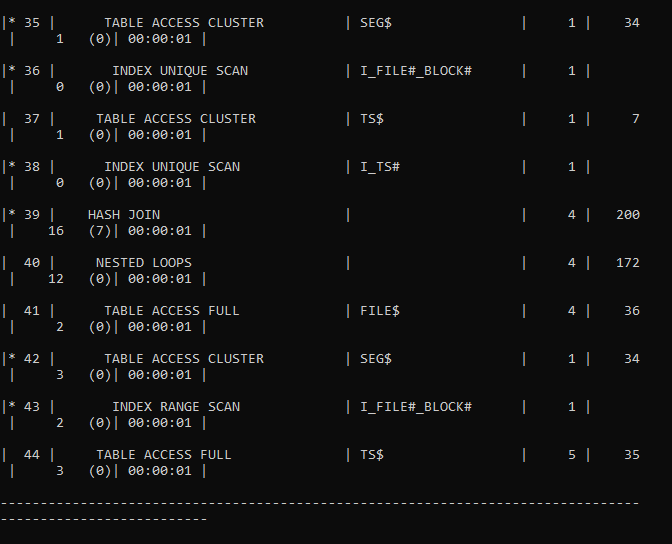
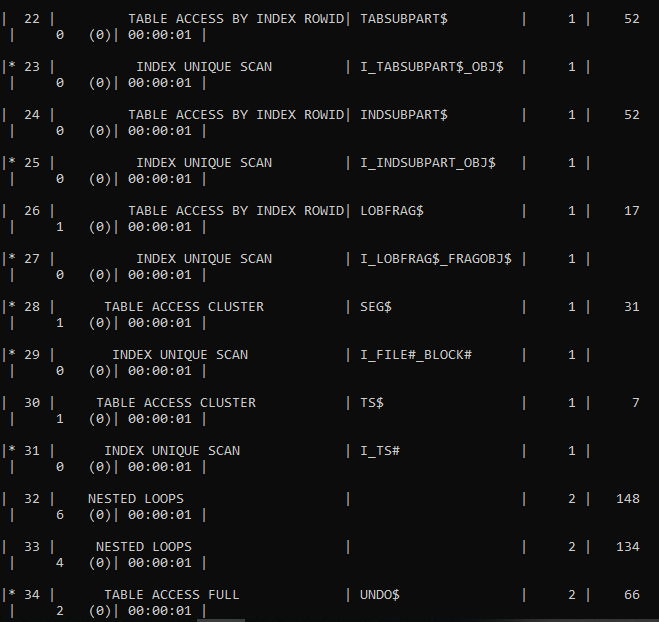
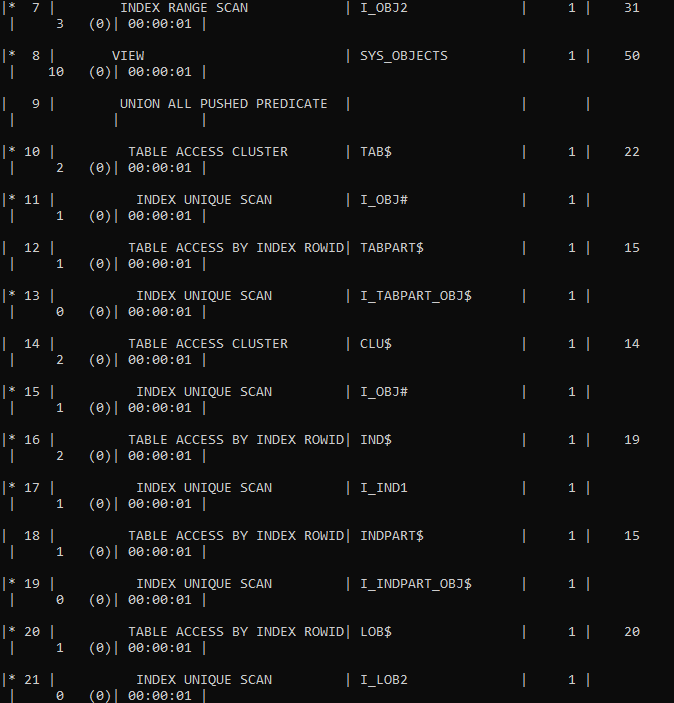
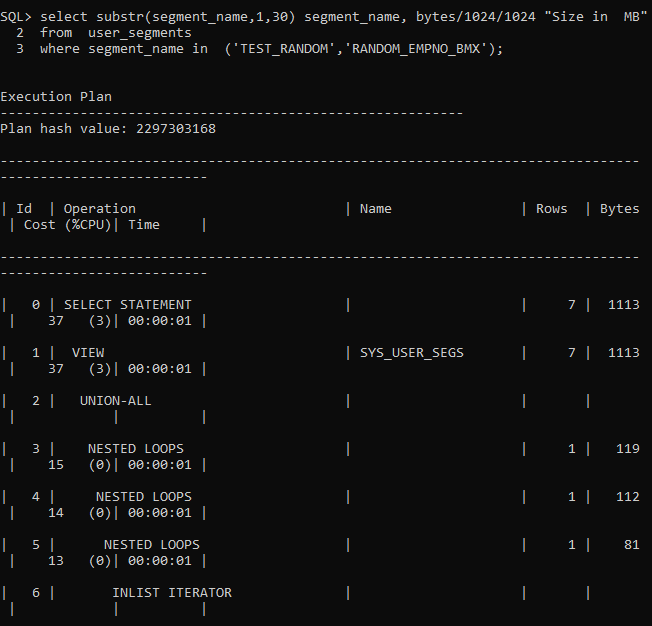
****

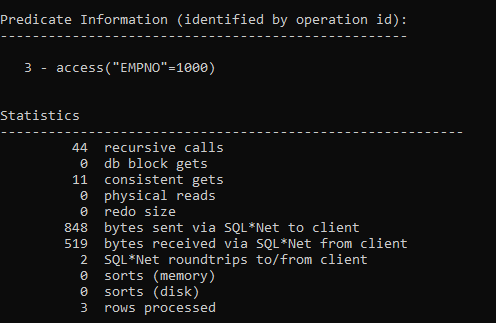
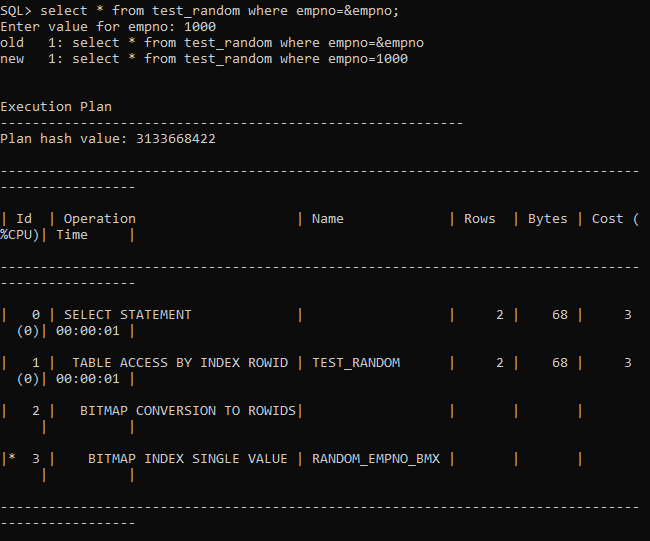
****

**Paso 2A (con TEST\_RANDOM)**

Ahora realizaremos el mismo experimento con TEST\_RANDOM

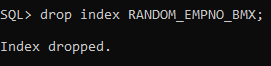
****

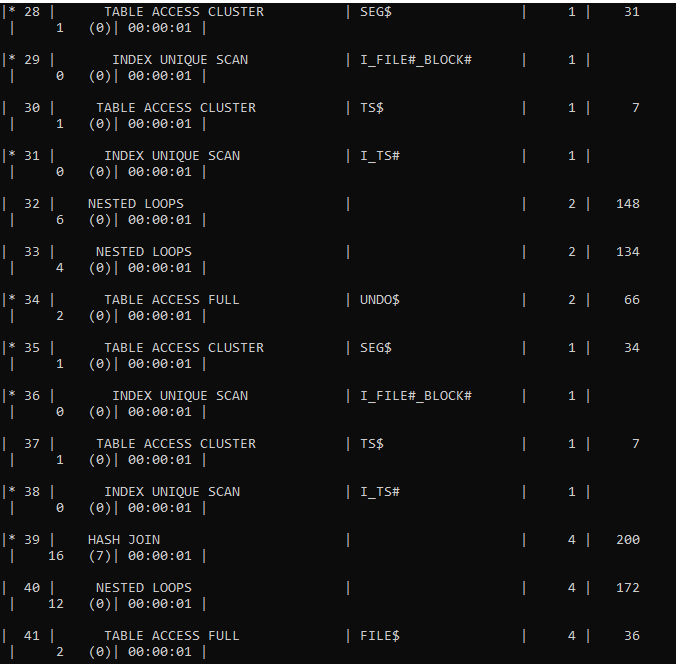
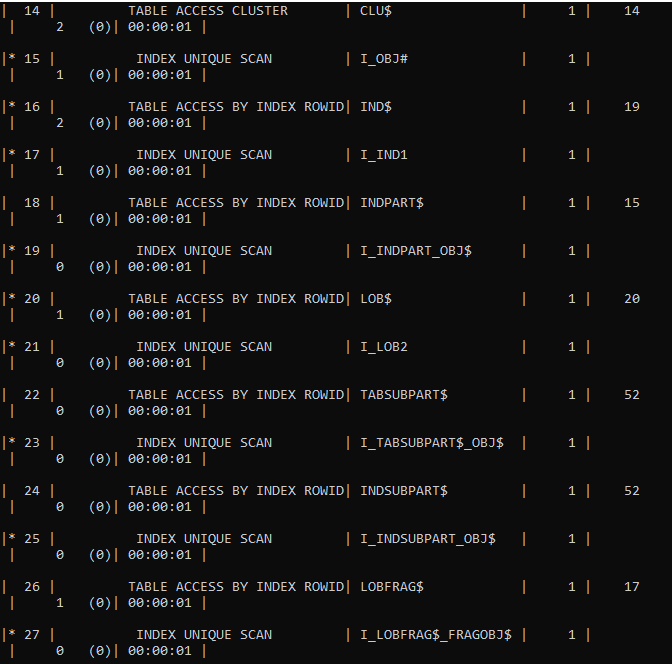
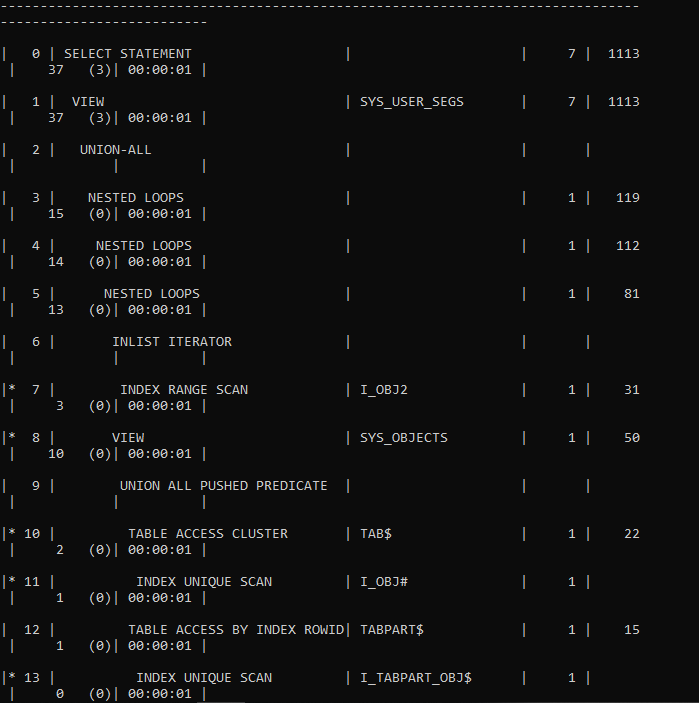
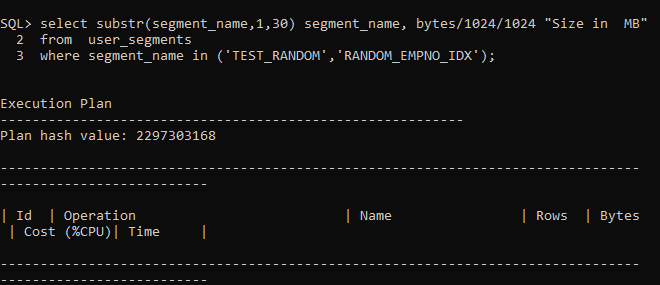
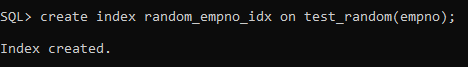
****

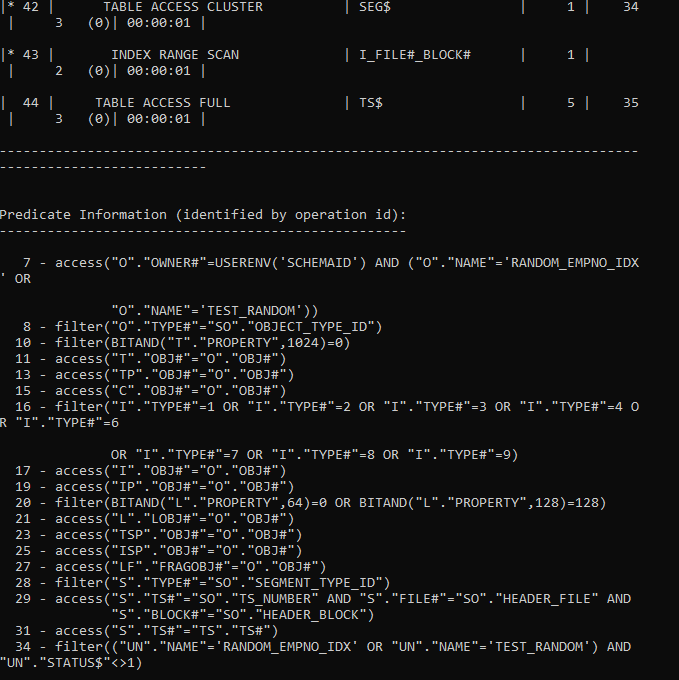
****

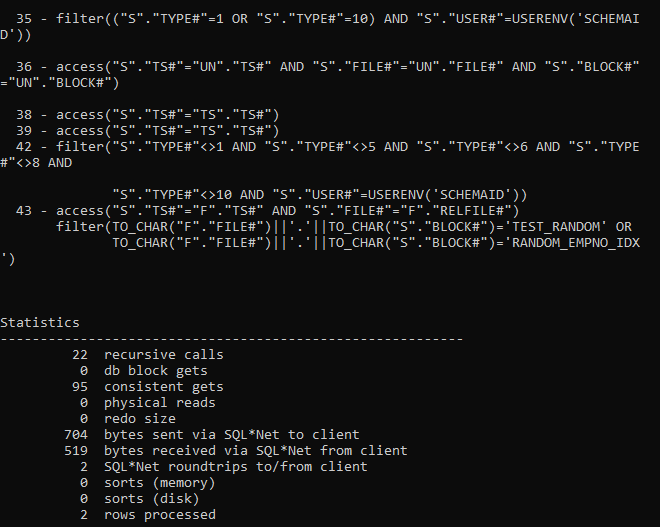
**Paso 2B (con TEST\_RANDOM)**

Ahora, como en el paso 1B, quitaremos el índice de mapa de bits y crearemos un índice de árbol B asociado a la columna EMPNO.

****

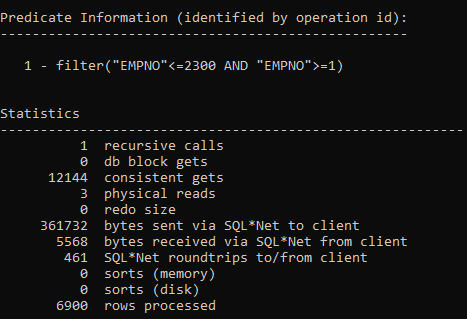
****

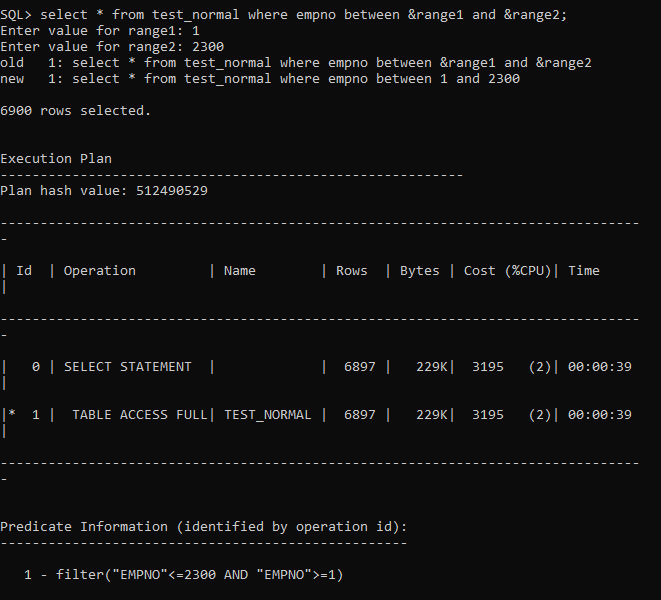
****

****

**Paso 3A (con TEST\_NORMAL)**

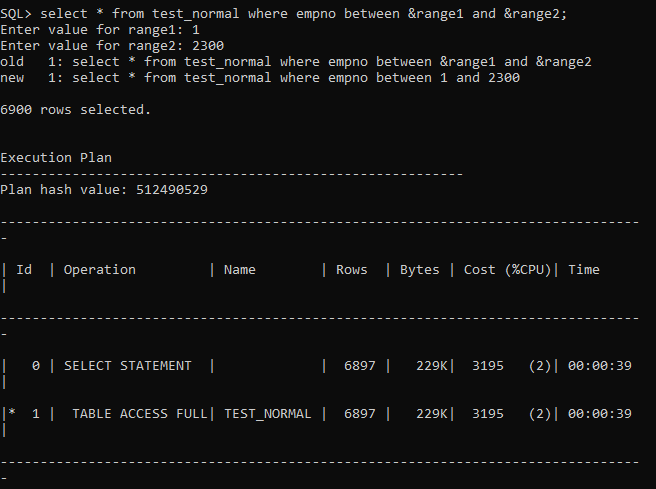
En este paso, crearemos el índice de mapa de bits (de manera similar a lo que hicimos en el paso 1A). Conocemos el tamaño del índice y el factor de agrupamiento del índice, que es igual a la cantidad de las filas de la tabla. Ahora ejecutaremos algunas consultas con predicados de rango.

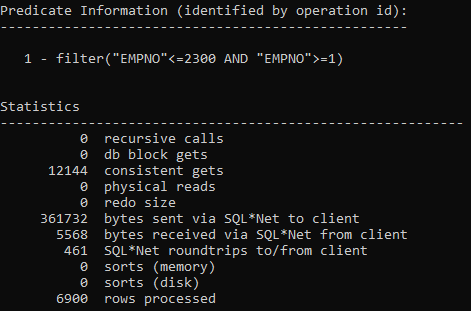
****

****

**Paso 3B (con TEST\_NORMAL)**

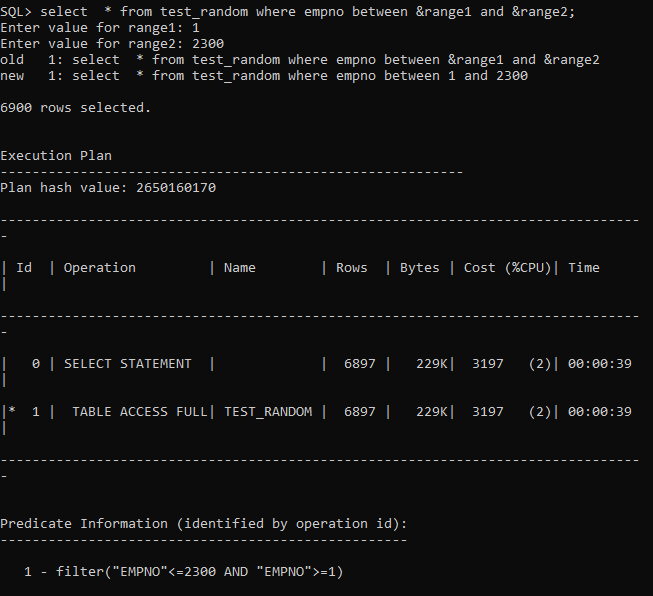
En este paso, ejecutaremos las consultas en la tabla TEST\_NORMAL con un índice de árbol B asociado a esa tabla.

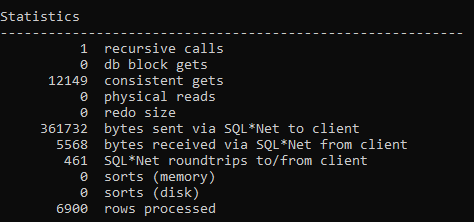
****



**Paso 4A (con TEST\_RANDOM)**

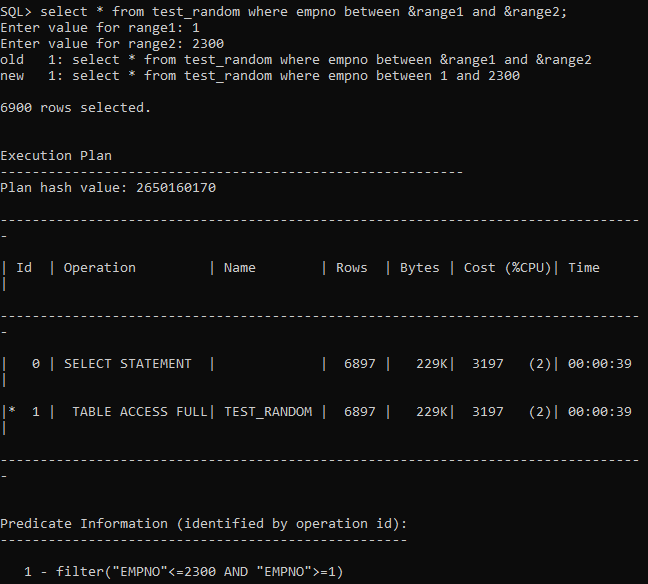
En este paso, ejecutaremos las consultas con predicados de rango en la tabla TEST\_RANDOM con el índice de mapa de bits y verificaremos la cantidad de lecturas coherentes y lecturas físicas. En este ejemplo se verá el impacto del factor de agrupamiento.

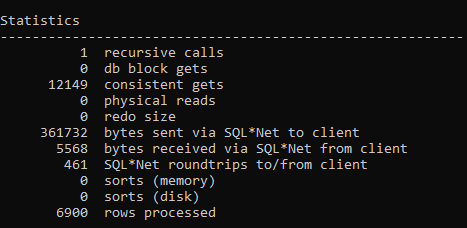




**Paso 4B (con TEST\_RANDOM)**

En este paso, ejecutaremos las consultas con predicados de rango en la tabla TEST\_RANDOM con un índice de árbol B asociado a esa tabla. Recuérdese que el factor de agrupamiento de este índice se aproximaba mucho a la cantidad de filas de la tabla (por lo que el índice resultaba ineficiente). A continuación, se incluye el procedimiento aplicado por el optimizador:

****

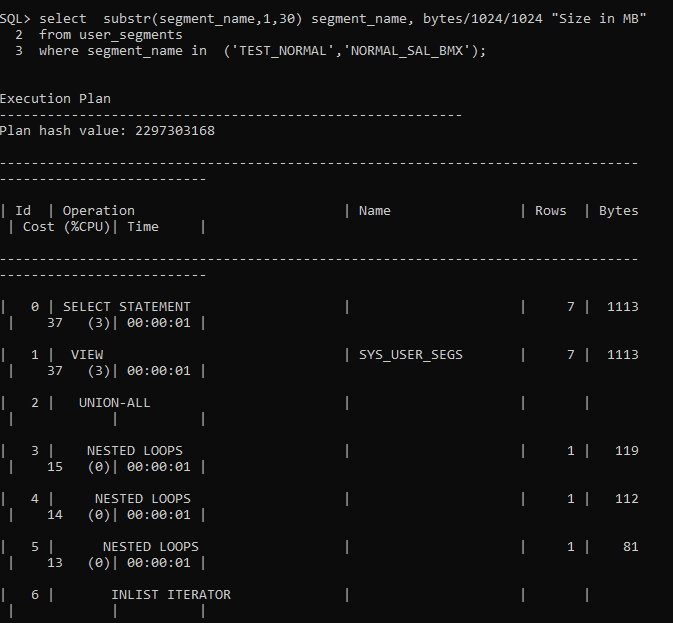
****

**Paso 5A (con TEST\_NORMAL)**

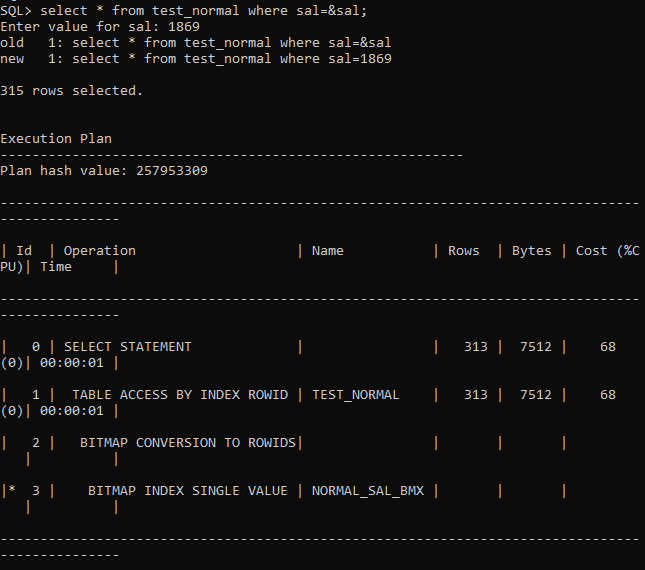
Crear un índice de mapa de bits asociado a la columna SAL [salario] de la tabla TEST\_NORMAL. Esa columna tiene cardinalidad normal.

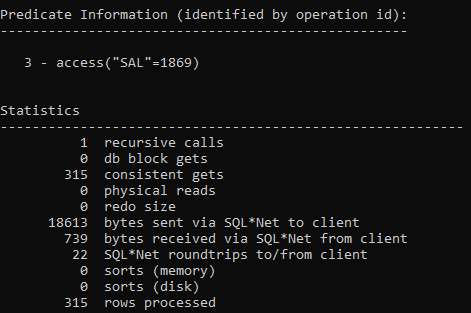
****

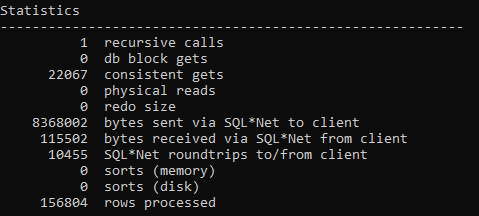
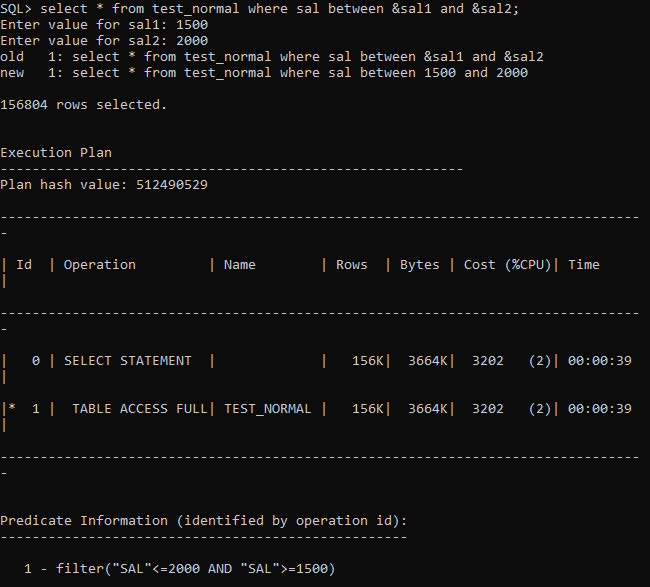
****

****

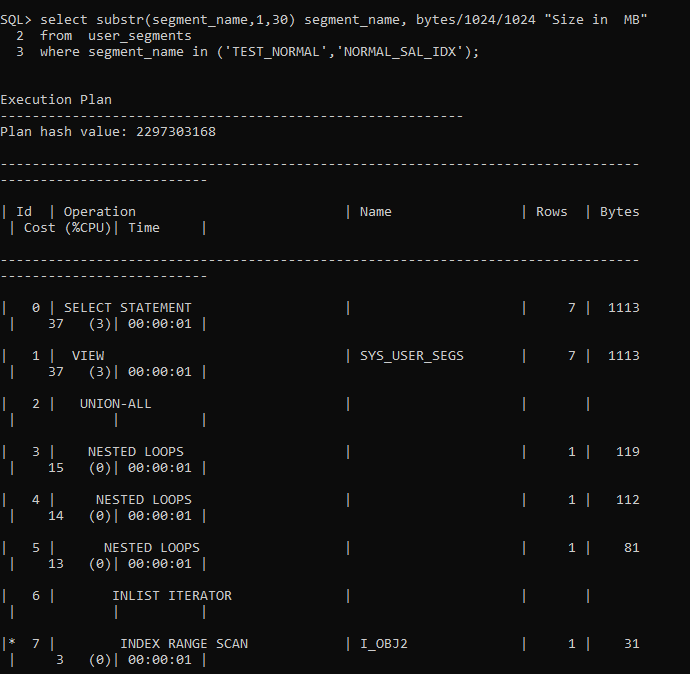
****

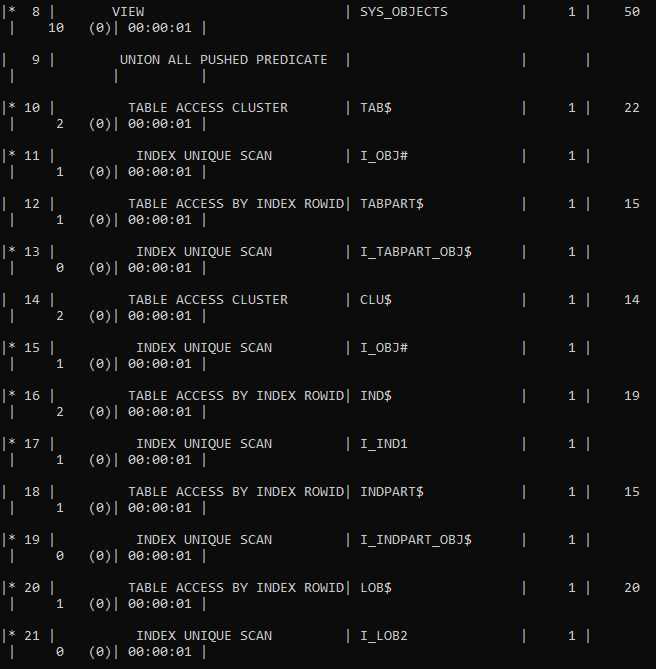
****

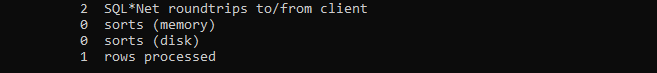
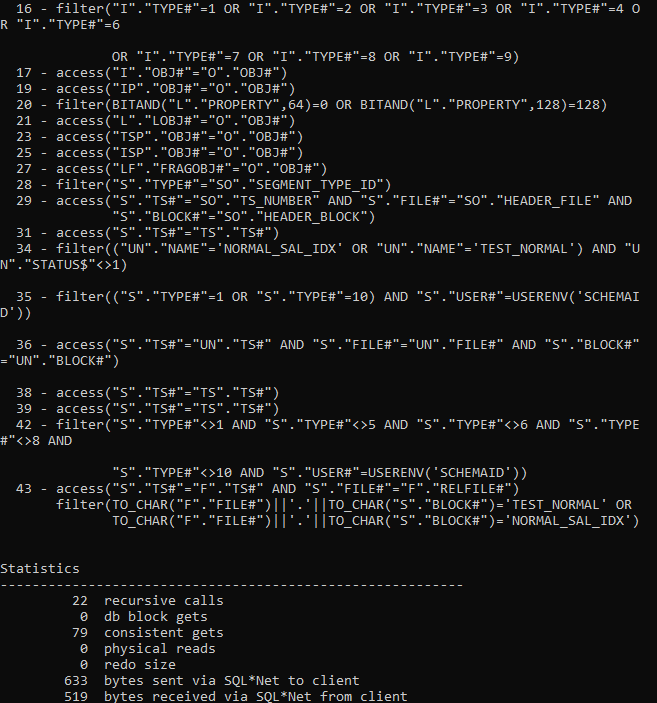
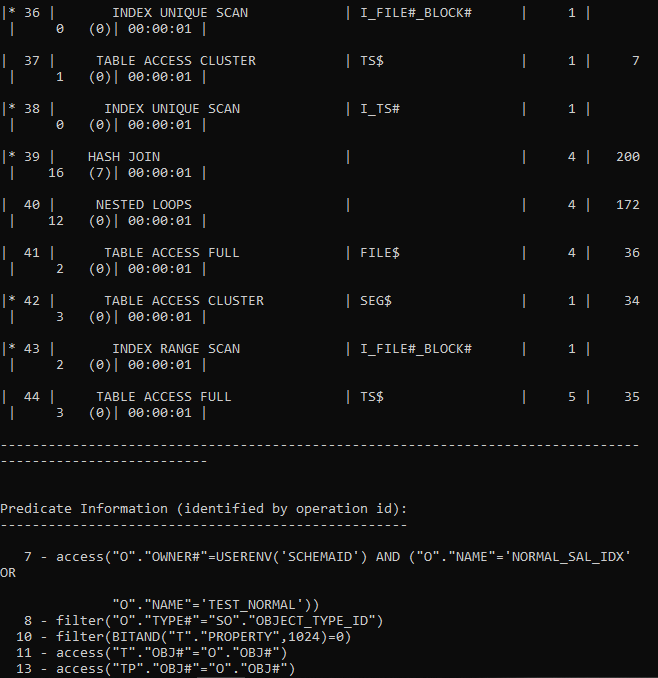
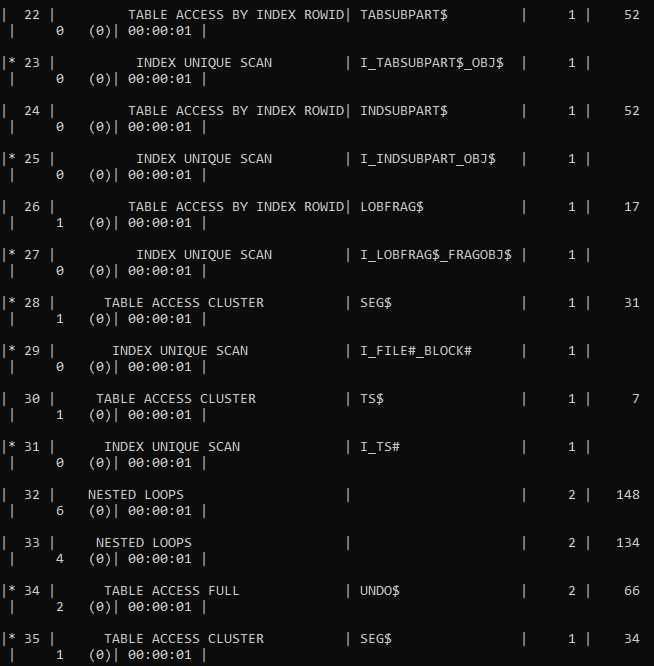
****

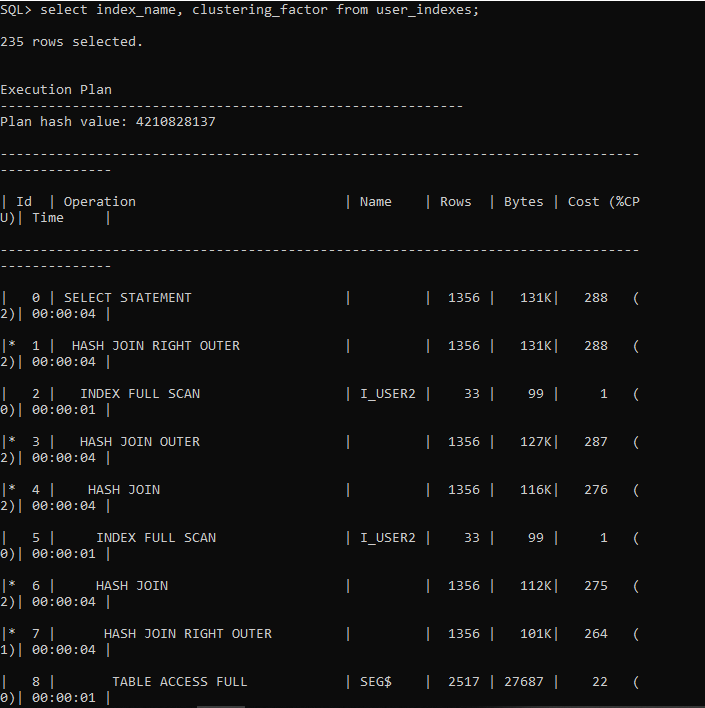
****

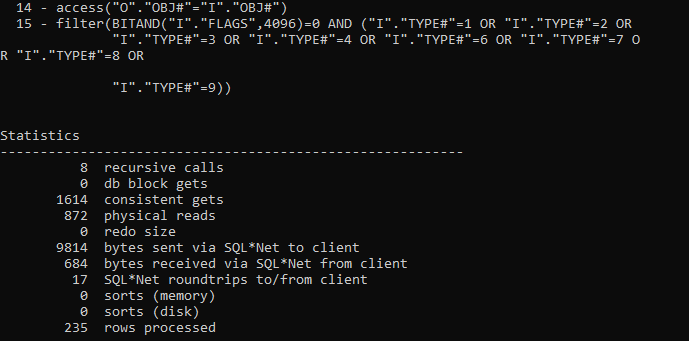
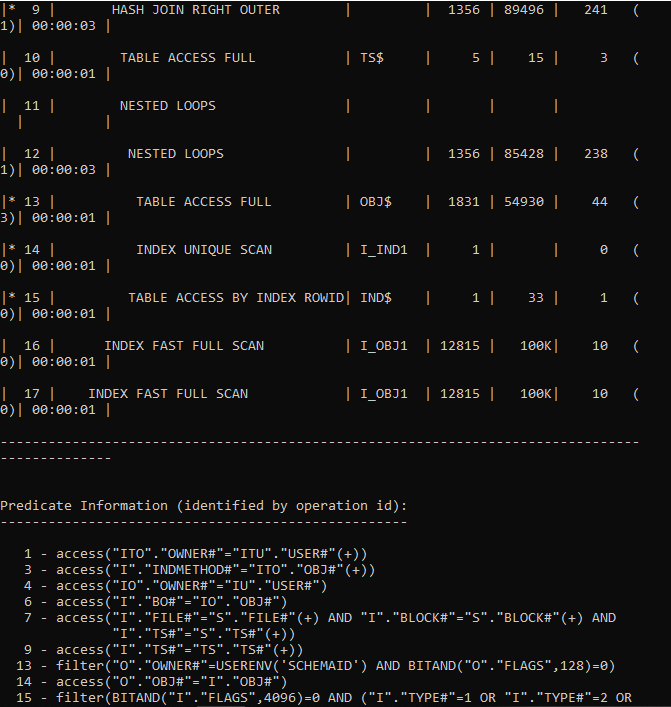
****

****

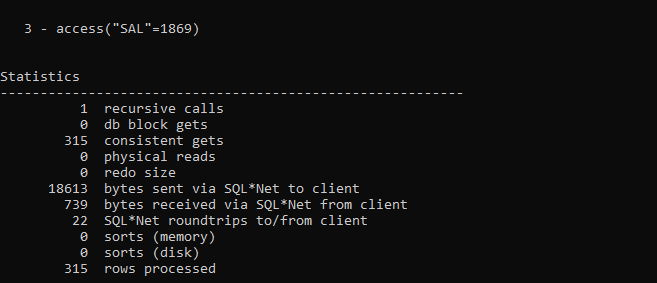
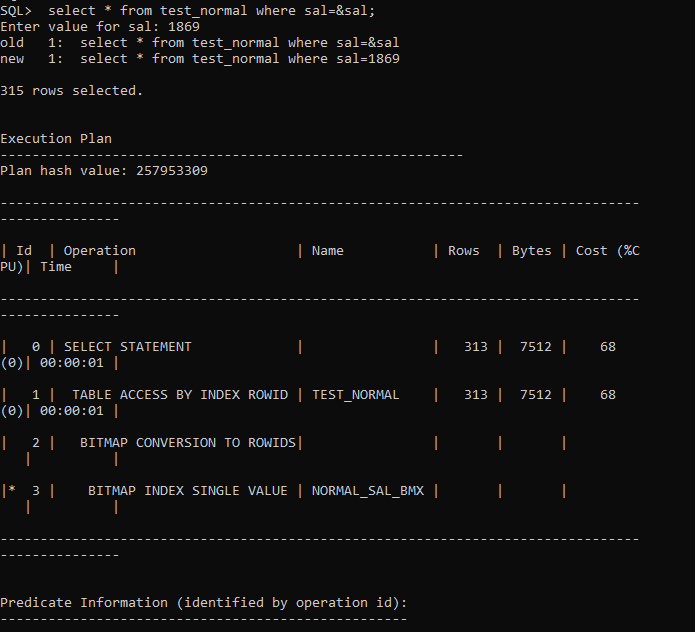
****

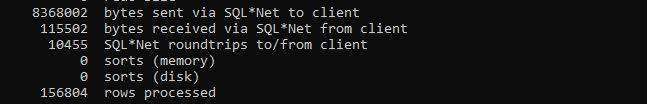
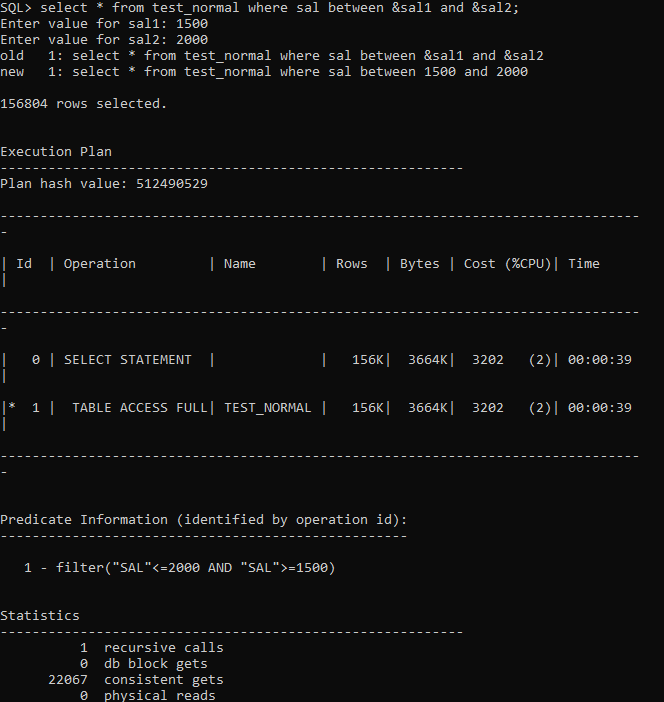
****

****

****

****

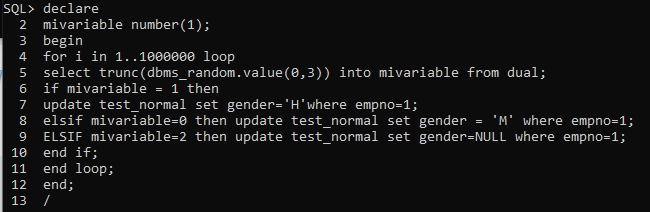
****

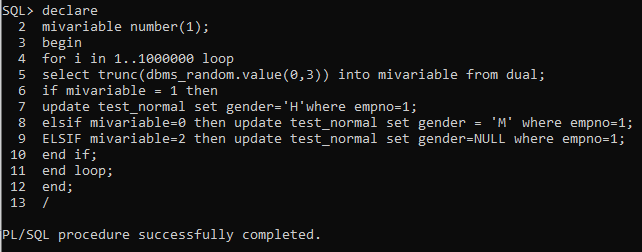
****

**Paso 6 (agregar una columna GENDER)**

Antes de realizar la prueba con una columna de baja cardinalidad, agregaremos una columna GENDER a la tabla y la actualizaremos para cargarle los valores *M*, *F* y *null*.

****

****

****