**DOCUMENTO ITERACIÓN 4**

**DISEÑO FÍSICO Y OPTIMIZACIÓN DE CONSULTAS**

Sistemas Transaccionales

Ingeniería de Sistemas y Computación

Universidad de los Andes

1. **Análisis:**

Para esta iteración no se realizó ninguna modificación a nivel del modelo del mundo.

1. **Diseño de la Aplicación:**

Para esta iteración no se realizó ningún cambio a nivel del modelo relacional.

1. **Diseño Físico:**

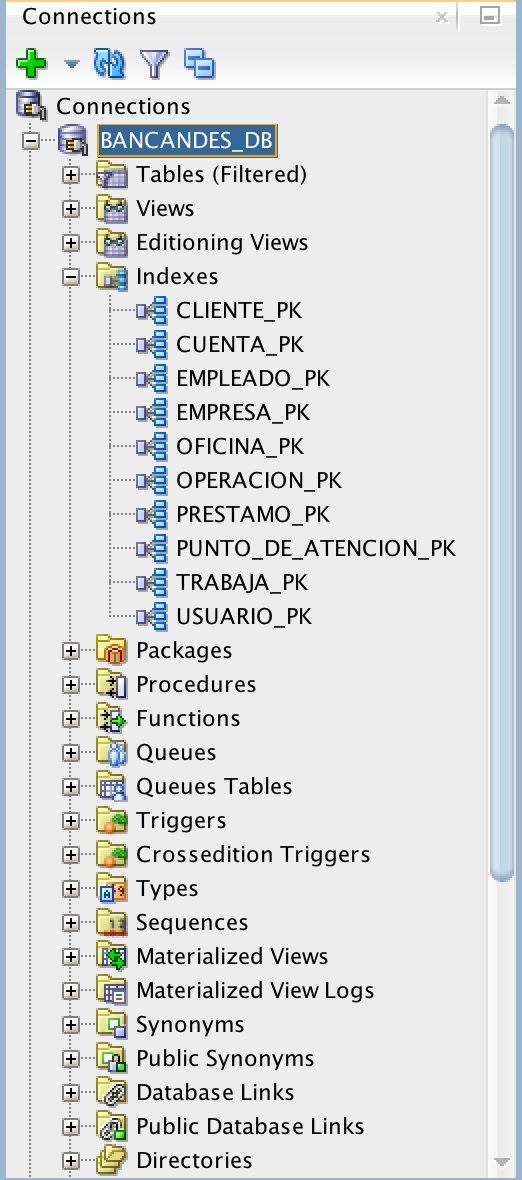
Un índice en una base de datos es una estructura que optimiza la velocidad de las operaciones y consultas, mediante el uso de identificadores en un tabla, permitiendo así un rápido acceso a los registros en una tabla. El espacio en memoria requerido para almacenar un índice es por lo general menor que el almacenamiento de la tabla. Se usan índices para aquellos valores sobre los cuales se hacen búsquedas frecuentes.

Así mismo existen condiciones que nos guían para saber cuando se deben usar índices. Cuando la selectividad está alrededor o menor de un 25% según Oracle, o cuando el WHERE se hace sobre una columna que tenga un rango, también sobre funciones agregadas (AVG, SUM, MAX) sin WHERE o GROUP BY, también cuando se usa ORDER BY sobre columnas que tengan un índice, y un CONSTRAINT de nulidad.

Por esta razón y para disminuir costos y tiempos en consulta se implementó el uso de índices en la base de datos de acuerdo a las consultas realizadas para satisfacer a cada requerimiento. A continuación se analiza cada requerimiento con una explicación de su proceso de desarrollo e implementación.

El costo mostrado en cada uno de los planes de consulta de Oracle / Requerimiento es un valor no determinado por operaciones de acceso a tablas. El valor no tiene ninguna unidad particular de medición. Es básicamente un valor usado para comparar costos de planes de ejecución. El valor es hallado a través de una función de CPU\_COST y IO\_COST.

A continuación esta la imagen de la información generada por Oracle sobre los índices antes de crearlos, ya que al trabajar sobre las mismas tablas para diferentes consultas no siempre los índices creados son buenos para todas las consultas y pueden afectar el resultado de otras.



Estos índices son creados por Oracle al ser PK con el fin de optimizar el tiempo de búsqueda y localización de los registros y hacerlos de una forma mas eficiente que si no implementara índices. Es este el principal factor para reducir I/O en el disco cuando son usados correctamente. Al ser atributos con valores distintos y no nulos, Oracle crea un índice automático.

1. **Requerimiento RFC7 - Consultar Operaciones V2**

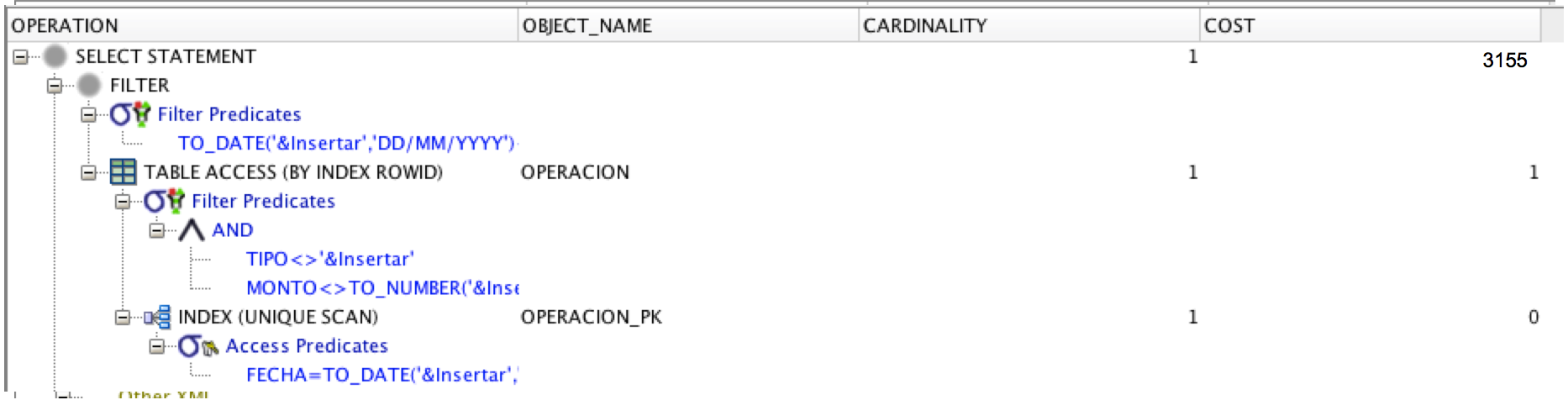
Dado un rango de fechas y un criterio de búsqueda asociado con la operación se obtienen los resultados de las operaciones realizadas

SELECT \* FROM OPERACION WHERE

FECHA<=TO\_DATE('&INSERTAR','DD/MM/YYYY') AND FECHA>=TO\_DATE('&INSERTAR','DD/MM/YYYY')

AND TIPO='&INSERTAR' AND MONTO='&INSERTAR';

Sin índices el tiempo de consulta par los datos es de:



Con la creación de los índices sobre los atributos:

CREATE INDEX INDEX\_OPERACION ON OPERACION(FECHA)

El tiempo de consulta es de:

**IMAGEN RESULTADOS INDEX**

Conclusiones:

* Luego de la creación de índices se logra una mejora de optimización, ya que la diferencia de tiempo de consulta es de (3155**-N**) lo cual es muy deseable, así mismo en el costo un número que halla el optimizador de Oracle basado en los I/O costos es menor también (3155 – 127).
* Los índices se crearon basados en los argumentos ya dichos anteriormente, principalmente por su porcentaje de selectividad, y su aporte en eficiencia de los JOINS usados en la consulta.

1. **Requerimiento RFC8 – Consultar Operaciones V3**

Dado un rango de fechas y un criterio de búsqueda asociado con la operación se obtienen los resultados de las operaciones realizadas fuera de este criterio de busqueda

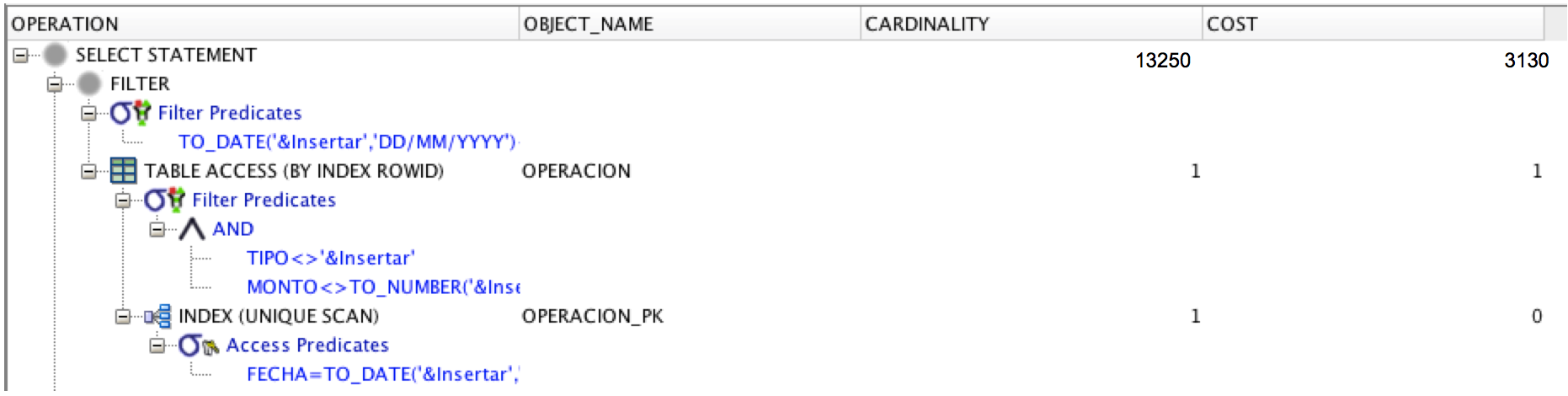
SELECT \* FROM OPERACION WHERE

FECHA<=TO\_DATE('&INSERTAR','DD/MM/YYYY') AND

FECHA>=TO\_DATE('&INSERTAR','DD/MM/YYYY')

AND TIPO!='&INSERTAR' AND MONTO!='&INSERTAR';

Sin índices el tiempo de consulta para los datos es de:

****

Con la creación de los índices sobre los atributos:

CREATE INDEX INDEX\_OPERACION ON OPERACION(FECHA)

El tiempo de consulta es de:

**IMAGEN RESULTADOS INDEX**

Conclusiones:

* Luego de la creación de índices se logra una mejora de optimización, ya que la diferencia de tiempo de consulta es de (3130**-N**) lo cual es muy deseable.
* Los índices se crearon basados en los argumentos ya dichos anteriormente, principalmente por su porcentaje de selectividad, y su aporte en eficiencia de los JOINS usados en la consulta.

1. **Requerimiento RFC9 – Consultar Consignaciones**

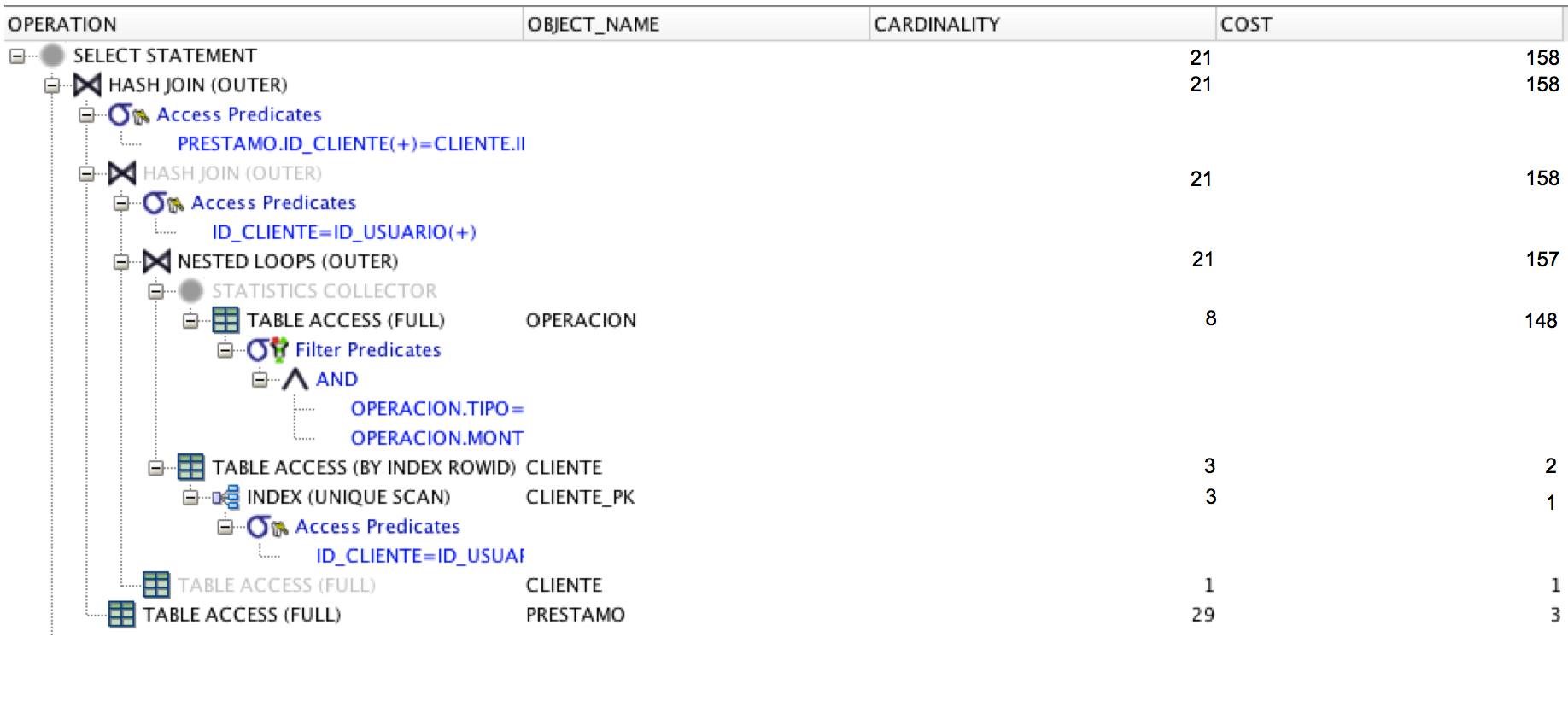
Dado un valor X y un tipo de préstamo Y se buscan las consignaciones con un monto mayor a X y préstamo tipo Y

SELECT \* FROM (OPERACION LEFT JOIN CLIENTE ON ID\_CLIENTE=ID\_USUARIO)

LEFT JOIN PRESTAMO ON PRESTAMO.ID\_CLIENTE=ID\_USUARIO

WHERE OPERACION.TIPO='CONSIGNAR' AND OPERACION.MONTO>='&INSERTAR';

Sin índices el tiempo de consulta para los datos es de:



Con la creación de los índices sobre los atributos:

CREATE INDEX INDEX\_CLIENTE ON CLIENTE(CEDULA)

CREATE INDEX INDEX\_PRESTAMO ON PRESTAMO(ID)

CREATE INDEX INDEX\_OPERACION ON OPERACION(FECHA)

El tiempo de consulta es de:

**IMAGEN RESULTADOS INDEX**

Conclusiones:

* Luego de la creación de índices se logra una mejora de optimización, ya que la diferencia de tiempo de consulta es de (158**-N**) lo cual es muy deseable, así mismo en el costo un número que halla el optimizador de Oracle basado en los I/O costos es menor también (3155 – 127).
* Los índices se crearon basados en los argumentos ya dichos anteriormente, principalmente por su porcentaje de selectividad, y su aporte en eficiencia de los JOINS usados en la consulta.

1. **Requerimiento RFC10 – Consultar Punto de Atención**

Dado el identificador de dos puntos de atención, mostrar la información de los clientes que han realizado operaciones en esos dos puntos de atención, incluyendo también la información de las operaciones realizadas.

1. **Análisis del proceso de optimización y el modelo de ejecución de consultas.**

* **JOIN**:

Cuando se lleva a cabo la operación de JOIN entre dos tablas con operaciones de control, es necesario comparar cada dato de la tabla A con cada dato de la tabla B. Este proceso requiere de leer el dato de cada tabla, realizar la comparación y escribir la respuesta. Dado que la respuesta es altamente variable con respecto al tamaño de las tablas, esta operación no se tiene en cuenta para calcular la complejidad.

El tiempo de realización del proceso, según el tamaño de la entrada entonces sería:

* **SELECT:**

Para realizar una operación de SELECT con un filtro solo es necesario tomar cada dato de la selección y evaluarlo según la condición de consulta. Esto es simplemente **O(n).**

* **ORDER BY:**

Para llevar a cabo una operación de ordenamiento en memoria principal, los algoritmos estables más eficientes tienen una complejidad de n\*log(n), y cargan todos los datos a memoria principal. Si los datos no caben en memoria principal, puede usarse *heap sort* que tiene la misma complejidad.

Oracle realiza operaciones similares, pero sobre bases de datos que no se encuentran en memoria principal, es por esto que si realizaran el mismo proceso donde se hace una lectura sobre el disco por cada dato para una consulta como la del requerimiento RFC7, donde la tabla OPERACIONES tiene entradas de un orden de 10^6 , el tiempo de ejecución sería de alrededor de 20 mil segundos o aproximadamente 5 horas y media.

Sin embargo cuando se ejecuta en el servidor Oracle, esta misma consulta tarda menos de 3 segundos sin usar índices, y un poco más de una décima de segundo al usar índices. Esto se debe a que la tarea del optimizador de consultas de Oracle usa herramientas como la clusterización de los datos, índices y características específicas de los datos para realizar el plan de ejecución de la manera más eficiente y no utilizar la totalidad de las tablas. De esta forma para realizar las operaciones solo requiere algunos datos que debido a las características de ordenamiento, son los que se sabe que deben ser evaluados.

La implementación de un optimizador de consultas es un proceso bastante complejo debido al gran numero de factores que se debe de tener en cuenta. En definitiva no es una función fácil de implementar y no sería rentable para la empresa desarrollarlo, es por esta razón que por términos de costos a nivel operativo es mas rentable usar los servicios de Oracle.