Sebastián Valencia Calderón.

Juan Camilo Bages.

**Análisis.**

Los requerimientos funcionales de la iteración anterior, contemplaban la inclusión de esquemas de concurrencia y de recuperación ante fallas, por lo que el análisis sobre los índices, se realizó sobre los requerimientos realizados sobre la presente entrega. Dada la metodología de desarrollo, la cual enfatiza el diseño de vistas por requerimientos, la implementación de los requerimientos pedidos para la presenta entrega, no intervienen en la implementación de los requerimientos pasados.

Los requerimientos para la presenta entregan se sintetizan a continuación, aquí se explica brevemente el requerimiento, su análisis para la implementación, aspectos de optimización y de diseño físico de la base de datos.

1. Consultar movimientos de valores (I).

Mostrar los movimientos de valores realizados en un rango de tiempo que corresponda a un criterio de búsqueda asociado con el movimiento, es decir, un atributo del valor o del movimiento en sí, este criterio es uno no más y es dado por el usuario. La implementación de este requerimiento, requiere la inclusión o adopción del concepto de solicitud dentro de la aplicación, para esto, es necesario incluir una relación sobre las tablas de solicitud y valor, esto se hace a partir de la asociación muchos a muchos presente en el modelo de siempre. La consulta requiere la revisión de las tablas de solicitud, portafolio, y usuario, ya que el criterio de selección se restringe a uno, es fácil implementar desde el punto de vista de aplicación la consulta. Esto es, sin tener en cuenta la implementación física y aspectos de optimización de la base de datos.

1. Consultar movimientos de valores (II).

Mostrar los movimientos de valores realizados en un rango de tiempo que corresponda a la negación de un criterio de búsqueda asociado con el movimiento, es decir, un atributo del valor o del movimiento en sí, este criterio es uno no más y es dado por el usuario. La implementación de este requerimiento, requiere la inclusión o adopción del concepto de solicitud dentro de la aplicación, para esto, es necesario incluir una relación sobre las tablas de solicitud y valor, esto se hace a partir de la asociación muchos a muchos presente en el modelo de siempre. La consulta requiere la revisión de las tablas de solicitud, portafolio, y usuario, ya que el criterio de selección se restringe a uno, es fácil implementar desde el punto de vista de aplicación la consulta. Esto es, sin tener en cuenta la implementación física y aspectos de optimización de la base de datos.

1. Consultar portafolios.

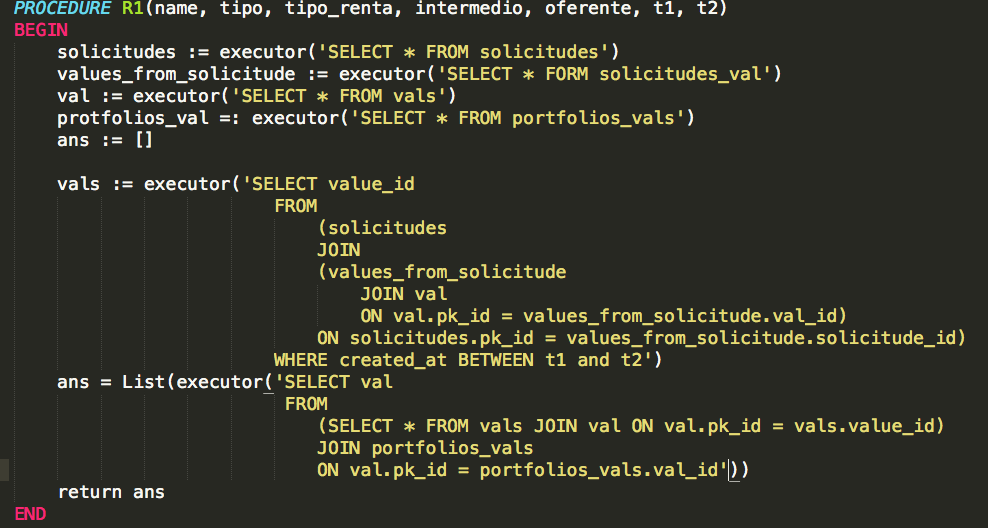
Éste requerimiento, es para mostrar los portafolios que contienen valores de un tipo X que han tenido operaciones con valor mayor a Y. Los criterios de búsqueda son dados por el usuario. La implementación de este requerimiento, no amerita la inclusión de nuevas tablas, sino la relación entre las tablas existentes. Estas tablas son: portafolio, valor, y transacción. La optimización requiere la revisión de los atributos que intervienen así como los de discriminación de datos.

1. Consultar valores (II).

Dado el identificador de un valor, mostrar la información de los portafolios en los que ha estado involucrado, incluyendo el periodo actual. Esta consulta requiere la revisión de las tablas valor, portafolio y solicitud.

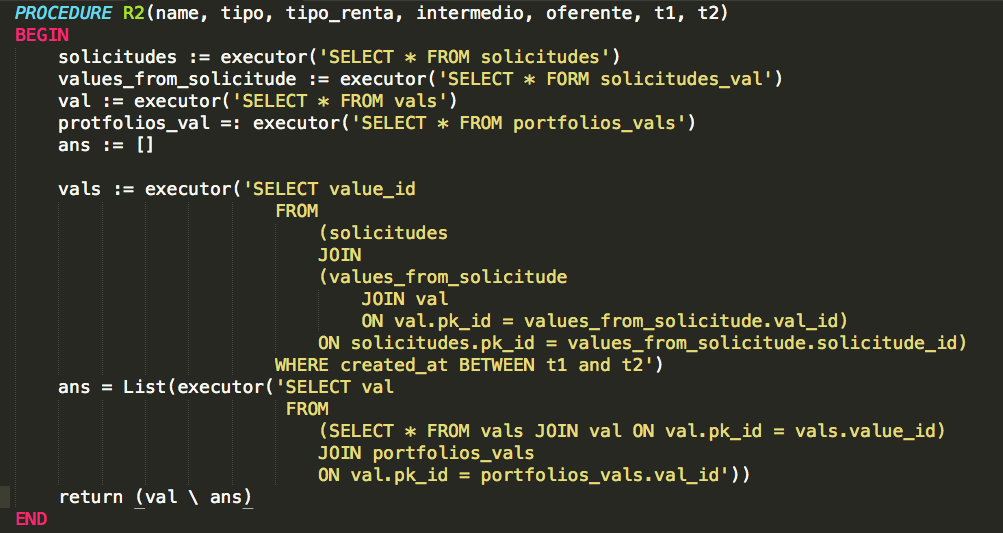
A continuación se muestra el análisis para la selección de índices, y el análisis sobre los índices existentes.

El esquema imperativo (Pascal) para el requerimiento 1 es el siguiente:



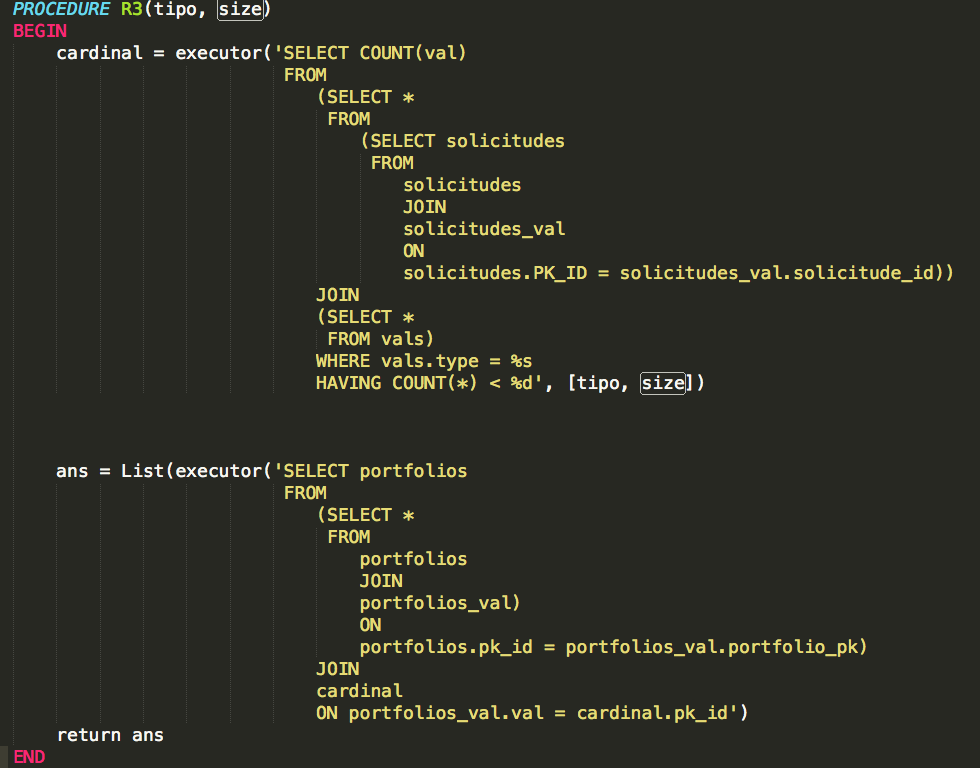
Los atributos candidatos para el índice son los involucrados en los JOIN, y en los criterios de selección del WHERE.

El esquema imperativo (Pascal) para el requerimiento 2 es el siguiente:



Los atributos candidatos para el índice son los involucrados en los JOIN, y en los criterios de selección del WHERE. Puede verse que el esquema es el mismo utilizado anteriormente pero el resultado se quita ans de los valores.

El esquema imperativo (Pascal) para el requerimiento 2 es el siguiente:



Los atributos candidatos para el índice son los involucrados en los JOIN, y en los criterios de selección del WHERE.

El último requerimiento, por el diseño del modelo relacional, involucra la selección de valores en el portafolio dado el tipo del valor. Los parámetros de tiempo, se estiman por las solicitudes.

Los criterios seleccionados para la selección de los índices son:

* Se justifica la creación de índices en columnas que son frecuentemente filtradas, es decir comúnmente usadas en la condición del WHERE.
* No se justifica la creación de índices sobre columnas que incurran a la aplicación de funciones de agregación frecuentemente, pues esto requiere la revisión de toda la tabla (excluyendo MAX, MIN).
* Se debe crear índices sobre columnas usadas en las condiciones de los JOIN.
* Se debe crear índices sobre columnas usadas como atributos de ordenamiento (ORDER BY).
* Se debe crear índices sobre columnas que tienen pocos valores repetidos o valores únicos en la tabla.
* No se debe crear índices sobre tablas de cardinalidad baja, pues el escaneo total puede resultar más efectivo que el QUERY indexado.
* Mantener en lo posible los índices cortos, el escaneo de índices agrupadas largos puede ser más caro que la búsqueda lineal.
* Crear índices sobre columnas con alta selectividad, es decir, columnas que no tengan muchas valores duplicados.
* Intentar crear índices agrupados sobre columnas que no serán frecuentemente actualizadas, cada vez que se actualice el índice, el motor de base de datos debe mantener este índice, por lo que no es buena idea crear índices sobre muchas columnas a la vez.
* No crear índices extra sobre las mismas columnas, pues además de no afectar el rendimiento de la ejecución, es espacio perdido y memoria invertida en el mantenimiento de los índices.

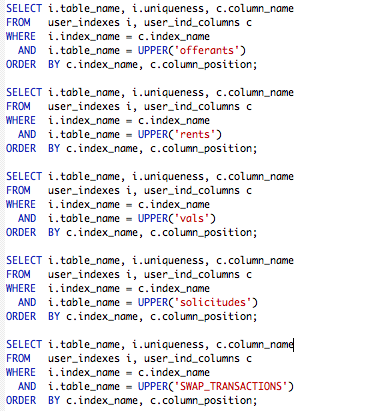
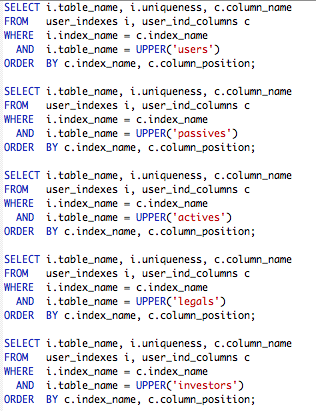
Los índices seleccionados y su justificación, se exponen a continuación:

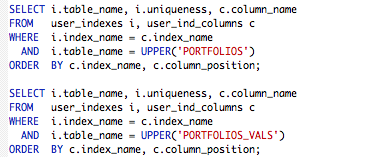
* Las llaves primarias de cada tabla: Son valores distintos y de baja actualización, además, son los fundamentales en la selección del JOIN.
* Las llaves foráneas de las tablas involucradas en el JOIN de selección: misma razón anterior.
* Las fechas, que son seleccionadas frecuentemente en los predicados de selección (WHERE), esto es siempre y cuando, la consulta no requiera la conversión de un tipo de dato, por ejemplo cadena de caracteres a fecha. Varios análisis sugieren la implementación de fechas a partir del UNIX timestamp por razones de eficiencia.
* Tipo de valor, de renta, son usados frecuentemente en las clausulas del WHERE.

Aparte de estos índices, Oracle indexa las siguientes columnas. Las tablas con columnas indexadas son las que tienen PK, esto favorece la implementación de los requerimientos por el diseño de la aplicación.

|  |  |
| --- | --- |
| INDEX\_NAME | TABLE\_NAME |
| AUTH\_GROUP\_PERMISSIONS\_0E9922A | AUTH\_GROUP\_PERMISSIONS |
| AUTH\_GROUP\_PERMISSIONS\_837A862 | AUTH\_GROUP\_PERMISSIONS |
| AUTH\_PERMISSION\_417F1B1C | AUTH\_PERMISSION |
| AUTH\_USER\_GROUPS\_0E939A4F | AUTH\_USER\_GROUPS |
| AUTH\_USER\_GROUPS\_E8701AD4 | AUTH\_USER\_GROUPS |
| AUTH\_USER\_USER\_PERMISSIONS1CCA | AUTH\_USER\_USER\_PERMISSIONS |
| AUTH\_USER\_USER\_PERMISSIONS8AFE | AUTH\_USER\_USER\_PERMISSIONS |
| D85D5DBB541FAB8F1DF26BA8B130BE | DJANGO\_CONTENT\_TYPE |
| DJANGO\_ADMIN\_LOG\_417F1B1C | DJANGO\_ADMIN\_LOG |
| DJANGO\_ADMIN\_LOG\_E8701AD4 | DJANGO\_ADMIN\_LOG |
| DJANGO\_SESSION\_DE54FA62 | DJANGO\_SESSION |
| INDEX3 | BEBEDORES |
| SYS\_C00135604 | USERS |
| SYS\_C00135606 | PASSIVES |
| SYS\_C00135607 | PASSIVES |
| SYS\_C00135611 | ACTIVES |
| SYS\_C00135617 | LEGALS |
| SYS\_C00135618 | LEGALS |
| SYS\_C00135621 | INVESTORS |
| SYS\_C00135624 | OFFERANTS |
| SYS\_C00135632 | RENTS |
| SYS\_C00135640 | VALS |
| SYS\_C00144877 | DJANGO\_MIGRATIONS |
| SYS\_C00144879 | DJANGO\_CONTENT\_TYPE |
| SYS\_C00144883 | AUTH\_PERMISSION |
| SYS\_C00144884 | AUTH\_PERMISSION |
| SYS\_C00144886 | AUTH\_GROUP |
| SYS\_C00144887 | AUTH\_GROUP |
| SYS\_C00144891 | AUTH\_GROUP\_PERMISSIONS |
| SYS\_C00144892 | AUTH\_GROUP\_PERMISSIONS |
| SYS\_C00144902 | AUTH\_USER |
| SYS\_C00144903 | AUTH\_USER |
| SYS\_C00144907 | AUTH\_USER\_GROUPS |
| SYS\_C00144908 | AUTH\_USER\_GROUPS |
| SYS\_C00144912 | AUTH\_USER\_USER\_PERMISSIONS |
| SYS\_C00144913 | AUTH\_USER\_USER\_PERMISSIONS |
| SYS\_C00144926 | DJANGO\_ADMIN\_LOG |
| SYS\_C00144931 | DJANGO\_SESSION |
| SYS\_C00145022 | SOLICITUDES |
| SYS\_C00145081 | SWAP\_TRANSACTIONS |
| SYS\_C00151606 | PORTFOLIOS |
| SYS\_C00151607 | PORTFOLIOS\_VALS |
| SYS\_IL0000159313C00003$$ | DJANGO\_ADMIN\_LOG |
| SYS\_IL0000159313C00006$$ | DJANGO\_ADMIN\_LOG |
| SYS\_IL0000159322C00002$$ | DJANGO\_SESSION |

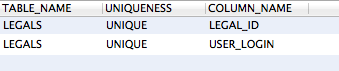
Las demás tablas indexadas corresponden a tablas de administración por parte del servidor de aplicaciones usador, estas son para la administración de sesión y demás. A continuación se muestran las tablas, con sus índices, así como las sentencias para obtener tal información. Claramente, los índices mostrados corresponden a las llaves primarias de cada tabla, pues sólo se muestran tablas con llave primaria, esto es por que es un dato de baja actualización, y además de alta dispersión sobre el dominio. No ayuda si la tabla posee pocos valores. Con respecto a los requerimientos funcionales, ayuda pues las llaves primarias son altamente usadas en la selectividad de los JOINS y de los predicados del WHERE.





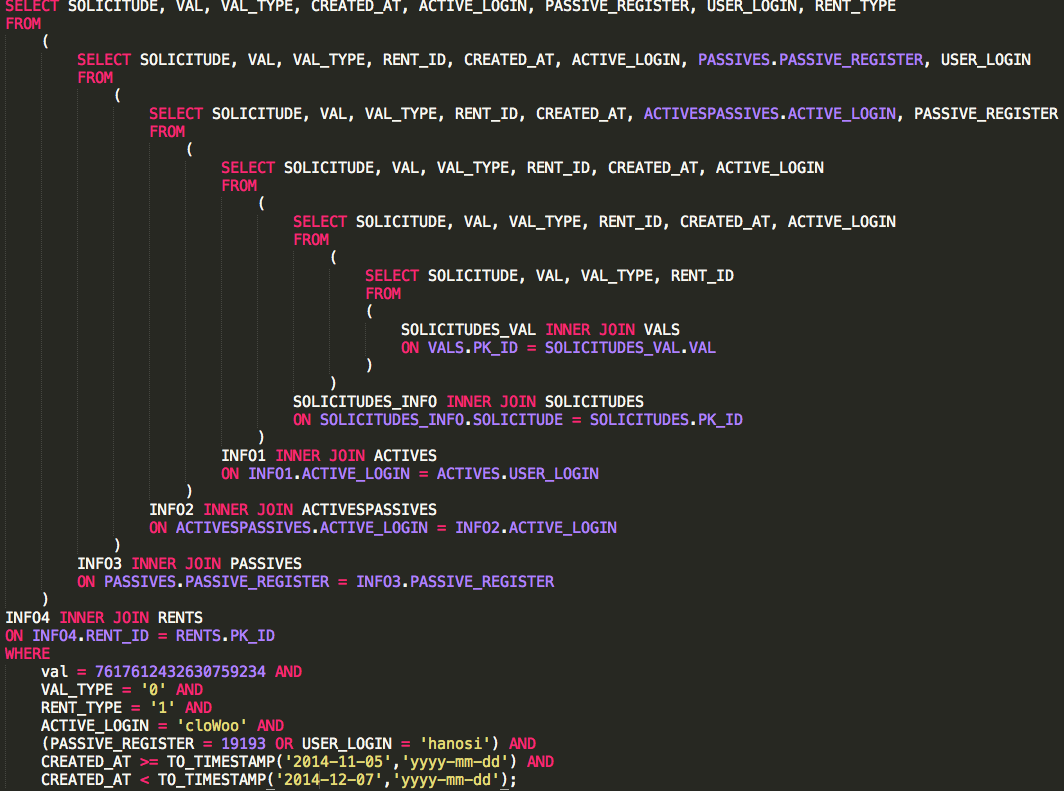




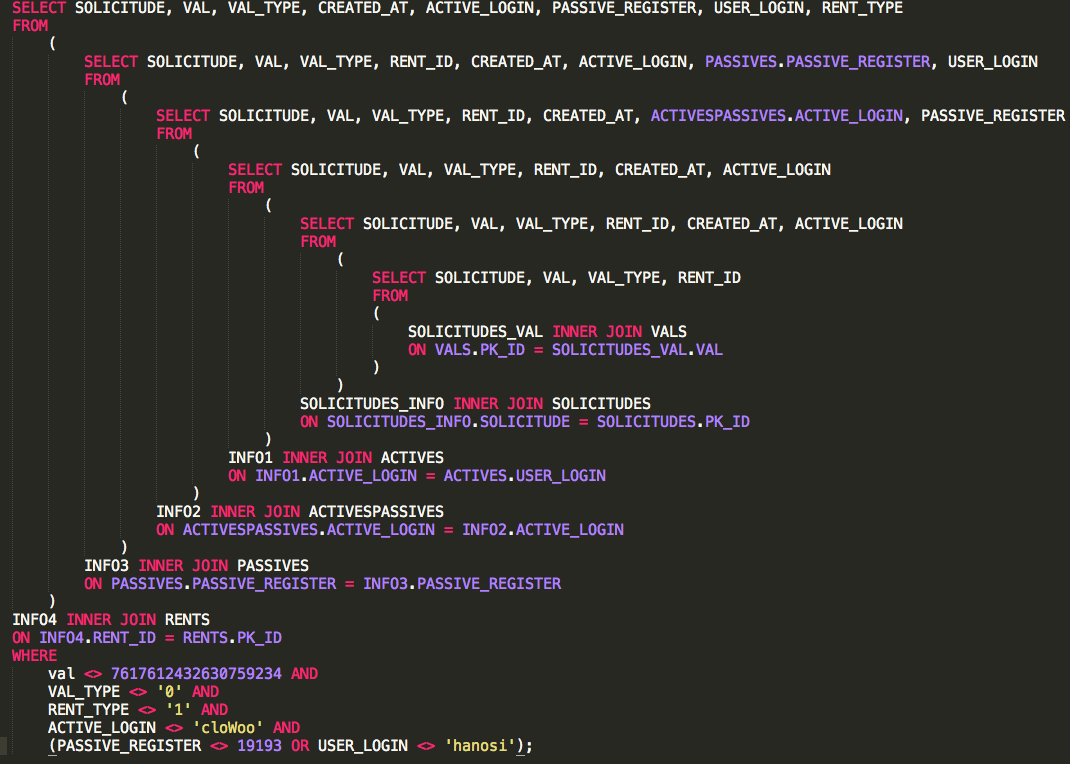


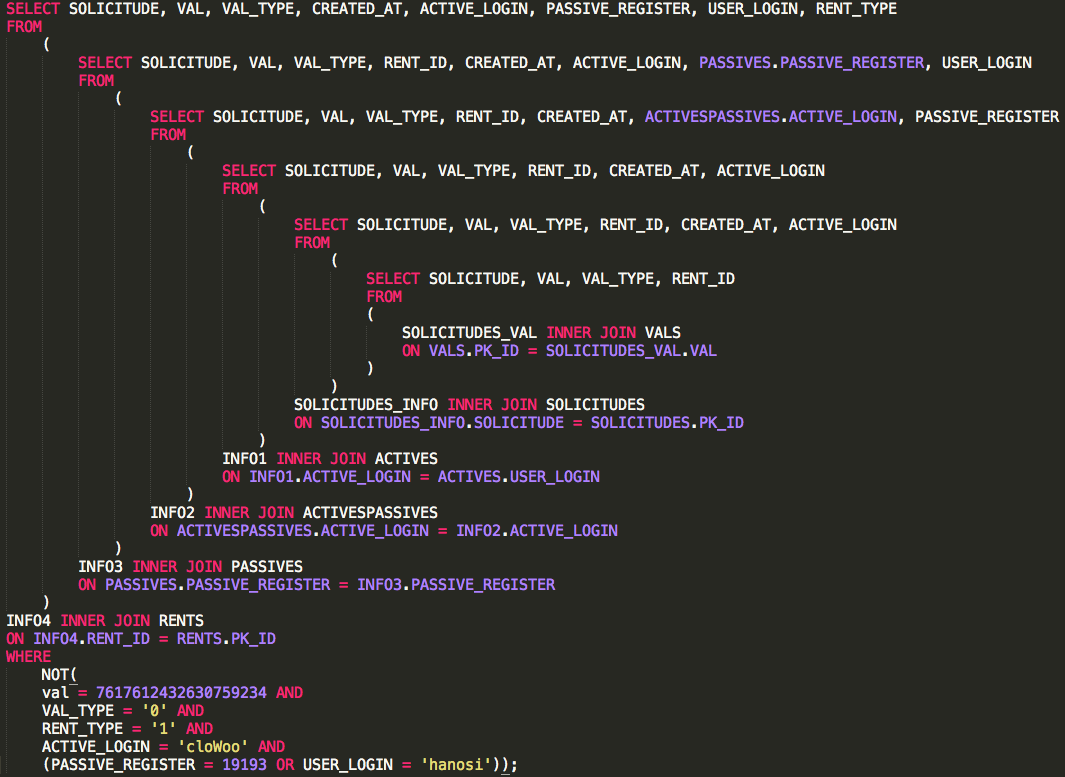
Las sentencias de ejecución para cada requerimiento son:

**Consultar movimientos valores (I):**

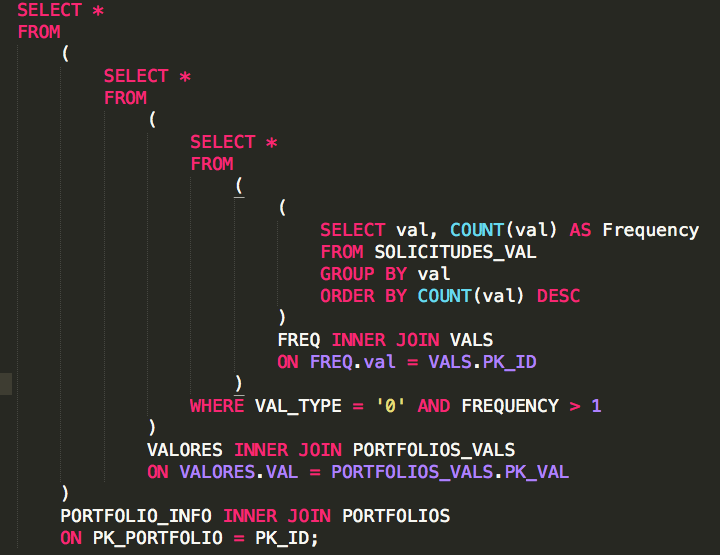
****

**Consultar movimientos valores (II):**

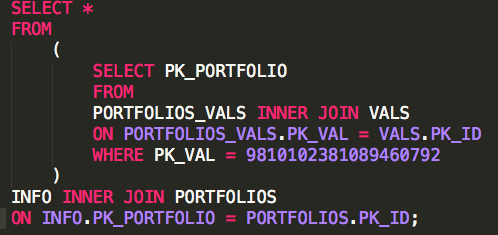
****

****

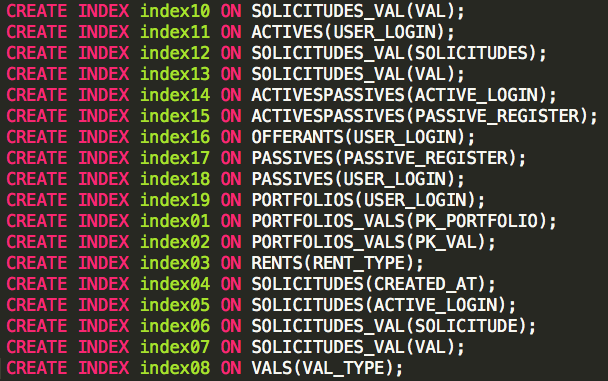
**Consultar portafolio:**



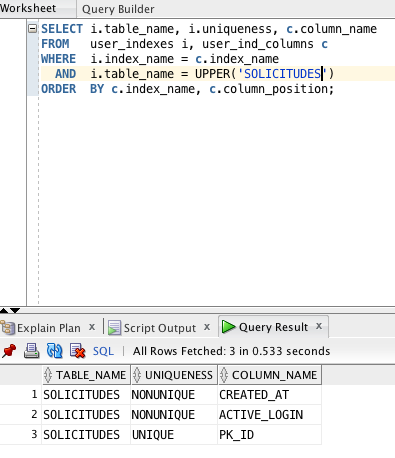
**Consultar valores 2:**

****

Los índices creados fueron:

****

Para mostrar la creación, se incluye la tabla con los índices sobre la tabla SOLICITUDES.

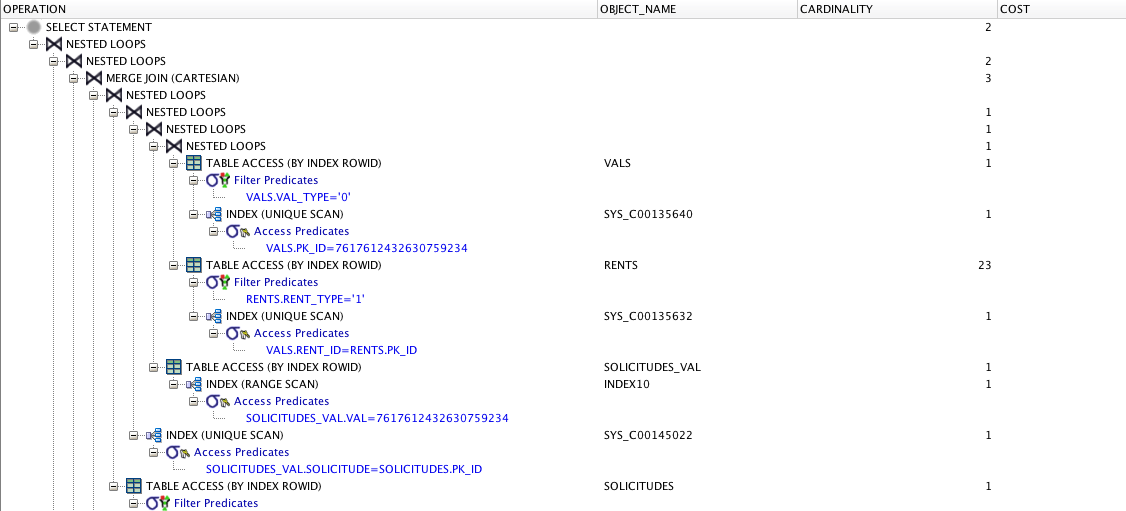
****

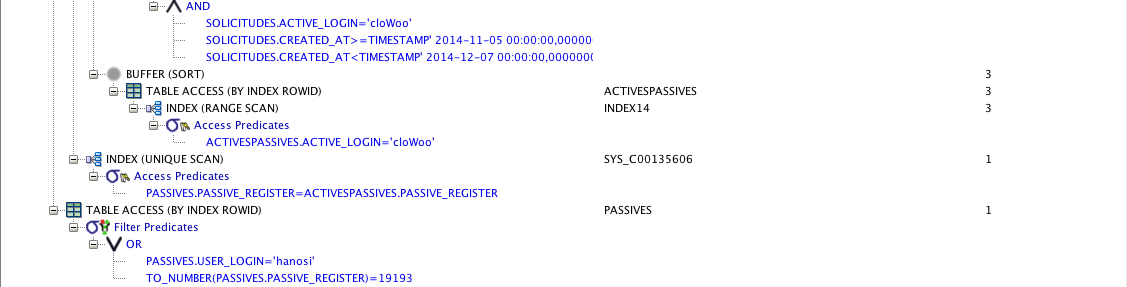
Los valores de prueba seleccionados se encuentran en las fotos de los comandos. Para llenar la base de datos, se usó un programa hecho en Python que selecciona valores aleatorios sobre grandes dominios, por lo que no hay datos diferenciadores sobre la ejecución.

El gráfico de distribución se muestra en el archivo adjunto DISTRIBUTION

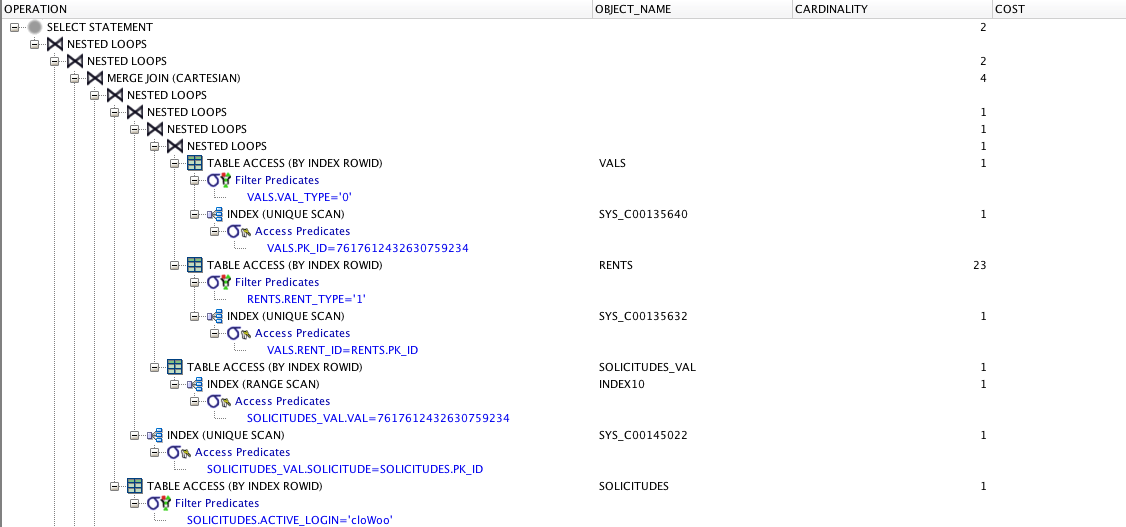
Los planes de consulta por requerimiento son los siguientes:

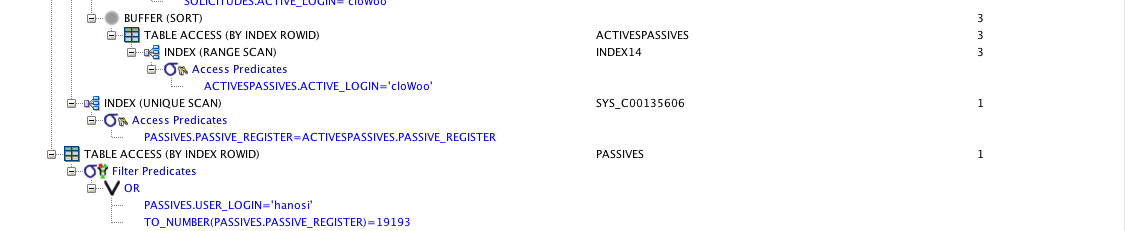
R1:



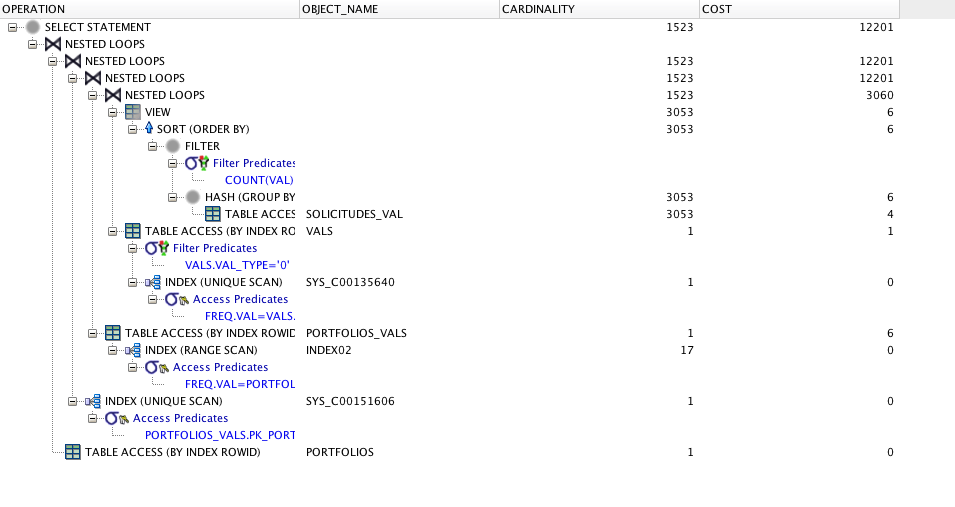


R2:

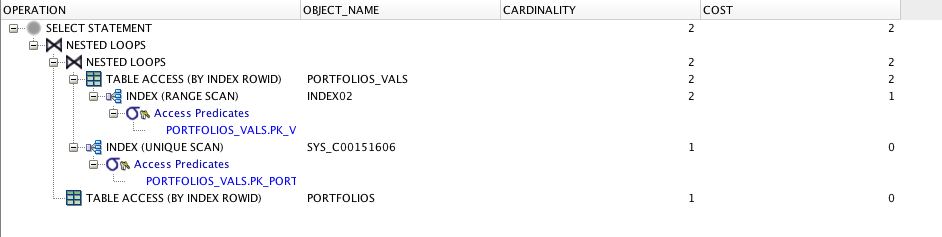




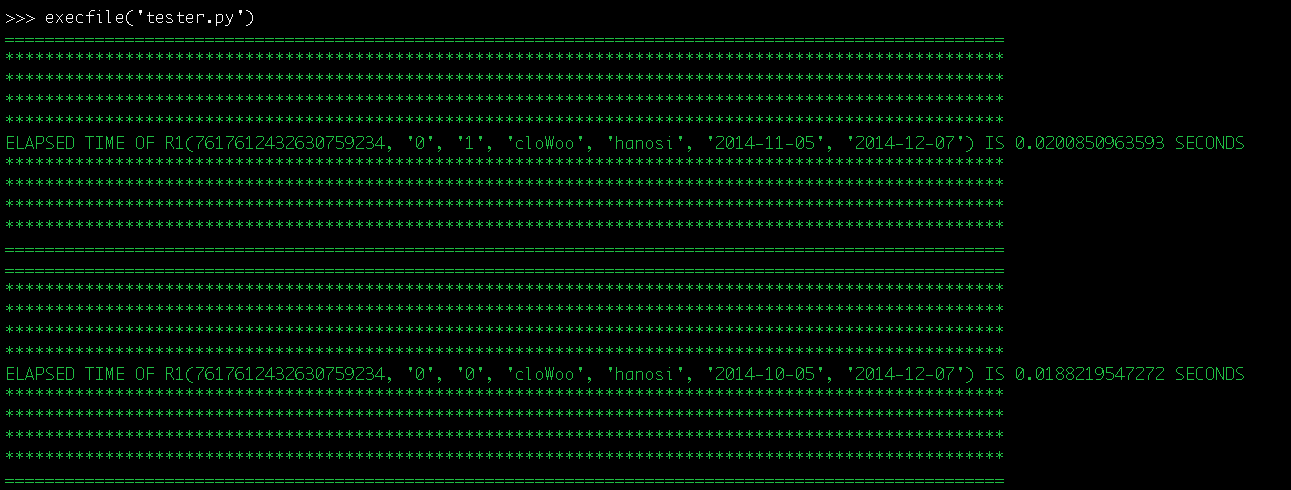
R3:

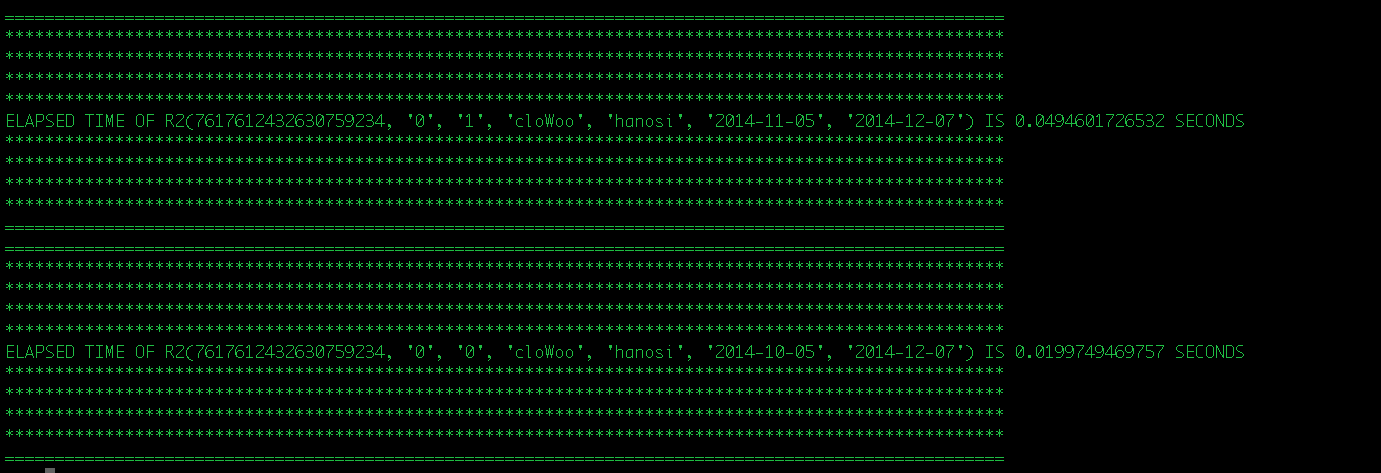


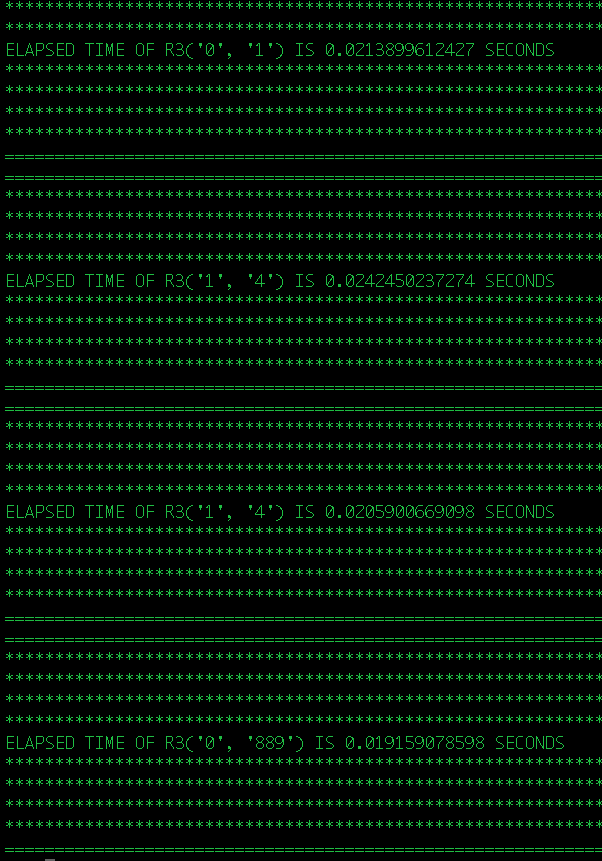
R4:

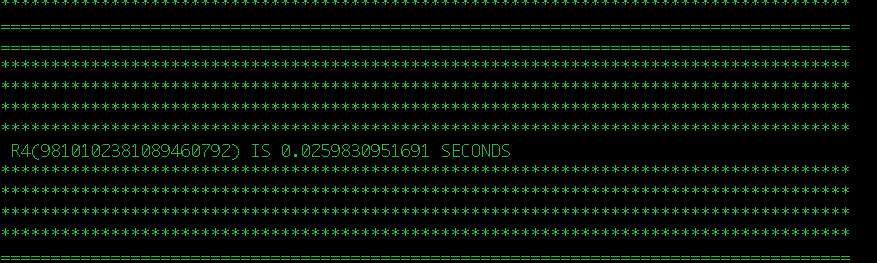


Los tiempos de ejecución son:



****

****

****

**R1:**

Alta selectividad por naturaleza del predicado.

Realizar JOIN usando loops anidados, y seleccionar usando IF en comparación, el de Oracle usa el INDEX.

**R2:**

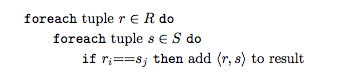
Igual que R1

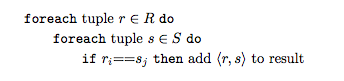
**R3:**

Loop anidado y selección sobre datos de índice, alta selectividad por cardinalidad de las tablas que intervienen.

**R4:**

Loop anidado y selección sobre datos de índice, alta selectividad por cardinalidad de las tablas que intervienen.





**Implementación.**

Se expone ahora de manera general un análisis comparativo de la discriminación de datos por parte de Oracle, y la discriminación de datos por parte del programador. El esquema de datos propuesto es el siguiente:

**Entrada**

|  |  |  |
| --- | --- | --- |
| **Titulo** | **Contenido** | **Fecha** |
| El cáncer de mama | El cáncer de mama es malo, muy muy malo | 24-10-2014 |
| Los atracos de transmilenio | Los atracos de transmilenio son malos, muy muy malos. | 22-10-2014 |
| Las drogas ilegales | Las drogas ilegales son malas, muy muy malas. | 20-10-2014 |
| Viajando por el mundo | Viajar por el mundo es bueno, muy muy bueno | 03-11-2014 |

**Comentario**

|  |  |  |
| --- | --- | --- |
| **Contenido** | **Entrada** | **Fecha** |
| De acuerdo | El cáncer de mama | 27-10-2014 |
| No de acuerdo | Los atracos de transmilenio | 24-10-2014 |
| Muy en contra | El cáncer de mama | 01-11-2014 |
| Mal mal mal | Viajando por el mundo | 06-11-2014 |

**Operaciones**

* **Selección (SELECT \* FROM Entrada WHERE Fecha > to\_date(‘15-10-2014’, ‘DD-MM-YYYY’)**

**Desarrollador:**

**Proceso:**

Para un desarrollador dando uso de un lenguaje de programación, la siguiente operación se realizaría con un recorrido sobre la tabla comparando los valores respectivos de fecha para saber si cumplen lo requerido y por ende son agregados al resultado.

**Complejidad:**

O(n) Donde n es la cantidad de datos en la tabla, esto es porque debe recorrer todos los elementos de la tabla al menos una vez.

**Base de datos:**

**Proceso:**

Para una base de datos, es posible realizar la creación de un índice en las fechas por lo que la cantidad de datos a recorrer disminuiría y de esta forma comparar cada fecha para saber si cumple el parámetro requerido.

**Complejidad:**

O(m) Donde m es la cantidad de distintas fechas en la tabla, la cual será considerablemente menor a la cantidad de datos en gran escala.

* **Join (SELECT \* FROM (Entrada INNER JOIN Comentario ON Titulo = Entrada) WHERE Entrada.Fecha > to\_date(26-10-2014))**

**Desarrollador:**

**Proceso:**

Para un desarrollador, es posible realizar esta operación mediante bucles anidados, en donde para cada dato válido encontrado en la tabla de Entrada, se realiza un recorrido sobre la tabla Comentario para obtener los datos correspondientes a esta tupla.

**Complejidad:**

O(nm) Donde n es la cantidad de datos en la primera tabla y m es la cantidad de datos en la segunda tabla.

**Base de datos:**

**Proceso:**

Busqueda lineal

**Complejidad:**

O(mnlog(mn))