

“Tercera Iteración Sistemas Transaccionales”

Mateo Devia, Felipe Velásquez
ISIS2304 - Ingeniería de Sistemas y Computación
Universidad de los Andes, Bogotá, Colombia
{m.devia, f.velasquez}@uniandes.edu.co
Fecha de presentación: Mayo 18 del 2018

Tabla de contenido

1	Introducción	1
2	Cambios en los Modelos	1
2.1	Cambios en el Modelo Conceptual.....	1
2.2	Cambios en el Modelo Relacional.....	2
3	Diseño Físico	8
3.1	Justificación de índices escogidos	8
3.2	Índices Generados automáticamente por	9
4	Documentación de los Requerimientos Funcionales y Análisis de eficiencia.....	9
4.1	Análisis de Eficiencia después de los índices y de las mejoras de eficiencia	10
5	Proceso de Carga de Datos	15
6	Análisis del proceso de optimización y el modelo de ejecución de consultas	16
7	Resultados	16
7.1	Logrados	16
7.2	No logrados.....	16
8	Conclusiones	16

1 Introducción

En el siguiente documento se exponen los resultados obtenidos tras el desarrollo del tercer proyecto o iteración de sistemas transaccionales. En él, se implementaron 4 nuevos requerimientos funcionales de manera eficiente. Para lograr dicha eficiencia se realizó un nuevo diseño de la base de datos con nuevos elementos como índices para poder lograr dicha eficiencia. Con esto se logró cumplir con los requerimientos no funcionales 6, 7, y 8 que exigen una cierta eficiencia en la aplicación. En el documento se documentarán todas las decisiones de diseño que se tomaron para el objetivo anteriormente mencionado, y se analizarán con respecto a las estrategias de eficiencia implementadas por Oracle.

2 Cambios en los Modelos

2.1 Cambios en el Modelo Conceptual

En cuanto al modelo conceptual se añadió un atributo nuevo a la entidad cliente llamado fecha de creación. Este atributo modela la fecha en que el cliente se registró en unidades. Este cambio se hizo con el propósito de poder identificar los buenos clientes de Alohandes (RFC13).

TIPO_DOCUMENTO	VARCHAR2	Tipo de documento (cédula, pasaporte, etc..)	NN, NC , CK, UA , ND1
NUM_DOCUMENTO	NUMBER	Número de documento del cliente	NN, NC , UA , ND1
TIPO_CLIENTE	VARCHAR2	Tipo de cliente (empleado, estudiante, etc..)	NN, CK, UA
FECHA_CREACION	DATE	Fecha en la que el cliente se registro en alohandes	NN, NC , UA

Tabla 1. Clientes

Nombre	PERSONAS_NAT		
Descripción	Guarda la información de las personas naturales que son operadores		
Llave	ID_OP		
Atributo	Tipo	Descripción	Características/Restricciones
ID_OP	NUMBER	ID del operador	PK, NN, ND, NC , SA FK(OPERDORES)
APELLIDO	VARCHAR2	Apellido de la persona	NN, UA
TIPO_DOCUMENTO	VARCHAR2	Tipo de documento de la persona	NN, NC , CK, UA
NUM_DOCUMENTO	NUMBER	Número de documento de la persona	NN, NC , UA
TIPO	VARCHAR2	Tipo de la persona (vecino o miembro de la comunidad)	NN, NC , CK, UA

Tabla 2. PersonasNat

Nombre	HABS_HOSTS		
Descripción	Contiene las habitaciones arrendadas dentro de casas de familia.		
Llave	ID_AL		
Atributo	Tipo	Descripción	Características/Restricciones
ID_AL	NUMBER	Indica el id del alojamiento a la que corresponde.	PK, NN, ND, NC , SA , FK(ALOJAMIENTOS)
NUM_COMP	NUMBER	Indica el número de personas con las que se comparte la habitación	NN, NC , UA

Tabla 3. HabsHosts

Nombre	OPERADORES		
Descripción	Tabla con todos los operadores de AlohAndes		
Llave	ID		
Atributo	Tipo	Descripción	Características/Restricciones
ID	NUMBER	ID del operador	PK, NN, ND, NC, SA
NOMBRE	VARCHAR2	Nombre del operador	NN, UA
TIPO	VARCHAR2	Tipo del operador (vecino, profesor, etc..)	NN, NC, CK, UA

Tabla 4. Operadores

Nombre	CASAS		
Descripción	Contiene los alojamientos de tipo casa		
Llave	ID_AL		
Atributo	Tipo	Descripción	Características/Restricciones
ID_AL	NUMBER	Id del alojamiento	PK, NN, ND, NC, SA , FK(ALOJAMIENTOS)
MENAJE	VARCHAR2	Descripción del menaje	UA
HABITACIONES	NUMBER	Número de habitaciones	NN, NC, UA
SEGURO	VARCHAR2	Descripción del seguro	UA

Tabla 5. Casas

Nombre	APARTAMENTOS		
Descripción	Tabla con los alojamientos de tipo apartamento		
Llave	ID_AL		
Atributo	Tipo	Descripción	Características/Restricciones
ID_AL	NUMBER	Identificador del alojamiento	PK, NN, ND, NC, SA , FK(ALOJAMIENTOS)

Tabla 6. Apartamentos

Nombre	VIVIENDAS_UNI		
Descripción	Contiene la información de las viviendas universitarias		
Llave	ID_OP		
Atributo	Tipo	Descripción	Características/Restricciones
ID_OP	NUMBER	Id del operador	PK, NN, ND, NC, SA , FK(OPERADORES)
DIRECCION	VARCHAR2	Dirección de la vivienda	NN, UA

Tabla 7. Viviendas universitarias

Nombre	HABS_VIVIENDAS		
--------	----------------	--	--

Descripción	Contiene las habitaciones ofrecidas por viviendas universitarias		
Llave	ID_AL		
Atributo	Tipo	Descripción	Características/Restricciones
ID_AL	NUMBER	Id del alojamiento	PK, NN, ND, NC , SA FK(ALOJAMIENTOS)
NUM_HAB	NUMBER	Número de la habitación	NN, NC , UA
NUM_COMP	NUMBER	Número de personas con las que se comparte la habitación	NN, UA

Tabla 8. Habitaciones de viviendas universitarias

Nombre	HOTELES		
Descripción	Contiene la información de los hoteles que prestan el servicio de alojamiento		
Llave	ID_OP		
Atributo	Tipo	Descripción	Características/Restricciones
ID_OP	NUMBER	Id del operador	PK, NN, ND, NC , SA FK(OPERADORES)
DIRECCION	NUMBER	Dirección del hotel	NN, UA
REGISTRO_CAMARA	VARCHAR2	Registro frente a la cámara de comercio.	NN, UA
REGISTRO_SUPER	VARCHAR2	Registro frente a la superintendencia.	NN, UA

Tabla 9. Hoteles

Nombre	HABS_HOTELES		
Descripción	Contiene la información de las habitaciones de hotel.		
Llave	ID_AL		
Atributo	Tipo	Descripción	Características/Restricciones
ID_AL	NUMBER	Id del alojamiento	PK, NN, ND, NC , SA FK(ALOJAMIENTOS)
NUM_HAB	NUMBER	Número de la habitación	NN, NC , UA
TIPO	VARCHAR2	Tipo de habitación (Suite, semisuite, etc..)	NN, CK, UA
TAMAÑO	NUMBER	Tamaño (en metros cuadrados) de la habitación	NN, UA

Tabla 10. Habitaciones de hoteles

Nombre	HOSTALES		
Descripción	Contiene la información de los hostales que ofrecen alojamientos.		
Llave	ID_OP		
Atributo	Tipo	Descripción	Características/Restricciones
ID_OP	NUMBER	Id del operador	PK, NN, ND, NC , SA , FK(OPERADORES)

HORARIO_APERTURA	VARCHAR2	Horario al que abre sus puertas	NN, UA
HORARIO_CIERRE	VARCHAR2	Horario al que abre sus puertas	NN, UA
DIRECCION	VARCHAR2	Dirección del hostel	NN, UA
REGISTRO_CAMARA	VARCHAR2	Registro frente a la cámara de comercio.	NN, UA
REGISTRO_SUPER	VARCHAR2	Registro frente a la superintendencia.	NN, UA

Tabla 11. Hostales

Nombre	HABS_HOSTALES		
Descripción	Contiene las habitaciones de los hostales		
Llave	ID_AL		
Atributo	Tipo	Descripción	Características/Restricciones
ID_AL	NUMBER	Id del alojamiento	PK, NN, ND, NC, SA FK(ALOJAMIENTO)
NUMERO	NUMBER	Número de la habitación	NN, NC, UA
NUM_COMP	NUMBER	Número de personas con las que se comparte la habitación	NN, UA

Tabla 12. Habitaciones de hostales

Nombre	ALOJAMIENTOS		
Descripción	Guarda la información general de todos los alojamientos		
Llave	ID		
Atributo	Tipo	Descripción	Características/Restricciones
ID	NUMBER	Id del alojamiento	PK, NN, ND, NC, SA
CAPACIDAD	NUMBER	Capacidad del alojamiento	NN, UA
COMPARTIDA	NUMBER	Indica si es compartido (1) o no (0)	NN, UA
TIPO	VARCHA2	Indica el tipo del alojamiento (habitación de hotel, de hostel, etc..)	NN, NC, CK, UA
UBICACIÓN	VARCHAR2	Describe la ubicación del alojamiento	NN, UA
ID_OP	NUMBER	Id del operador del alojamiento	NN, FK (OPERADORES)

Tabla 13. Alojamientos

Nombre	SERVICIOS		
Descripción	Contiene los servicios prestados por los varios alojamientos		
Llave	ID		

Atributo	Tipo	Descripción	Características/Restricciones
ID	NUMBER	Id del servicio	PK, NN, NC , SA
TIPO	VARCHAR2	Tipo del servicio (tv, recepción, etc...)	NN, NC , CK, UA
COSTO	DOUBLE PRECISION	Costo del servicio	NN, UA

Tabla 14. Servicios

Nombre	OFERTAS_ALOJAMIENTOS		
Descripción	Contiene las ofertas de alojamientos		
Llave	ID_AL, FECHA_CREACION		
Atributo	Tipo	Descripción	Características/Restricciones
PRECIO	DOUBLE PRECISION	Precio de la oferta	NN, UA
FECHA_RETIRO	DATE	Fecha en que se retirará la oferta	UA
UNIDAD_PRECIO	VARCHAR2	Unidad de precio de la oferta (dia, mes, semestre, etc...)	NN, CK, UA
ID_AL	NUMBER	Id del alojamiento al que corresponde la oferta	PK, NN, NC , SA FK(ALOJAMIENTOS)
FECHA_CREACION	DATE	Fecha de creación de la oferta	PK, NN, UA

Tabla 15. Ofertas de alojamiento

Nombre	RESERVAS		
Descripción	Contiene la información de las reservas		
Llave	FECHA_INICIO, ID_AL_OF, FECHA_CREACION_OF		
Atributo	Tipo	Descripción	Características/Restricciones
FECHA_INICIO	DATE	Fecha en que inicia la reserva	PK, NN, UA
FECHA_FIN	DATE	Fecha en que termina la reserva	NN, UA
TIPO_CONTRATO	VARCHAR2	Tipo de contrato de la reserva	NN, NC , UA
NUM_PERSONAS	INTEGER	Número de personas de la reserva	NN, UA
PRECIO_RESERVA	DOUBLE PRECISION	Precio acordado por el alojamiento	NN, UA
ID_CLIENTE	NUMBER	Id de quién hace la reserva	NN, NC , SA , FK(CLIENTES)
ID_AL_OF	NUMBER	Id de la oferta de alojamiento	PK, NN, NC , SA , FK (OFERTAS_ALOJAMIENTO)
FECHA_CREACION_OF	DATE	Fecha de creación de la oferta de alojamiento	PK, NN, NC , UA , FK(OFERTAS_ALOJAMIENTO)
ESTADO	VARCHAR2	Estado de la reserva (cancelada, reservada)	NN,CK, NC , SA

ID_COLECTIVA	NUMBER	Id de la reserva colectiva a la que pertenece NC, SA, FK(RESERVAS_COLECTIVAS)
--------------	--------	--

Tabla 16. Reservas

Nombre	ALOJ_SERV		
Descripción	Tabla que relaciona las ofertas con los servicios		
Llave	ID_ALOJ, ID_SERV		
Atributo	Tipo	Descripción	Características/Restricciones
ID_ALOJ	NUMBER	Id del alojamiento	PK, NN, UA , FK(ALOJAMIENTOS)
ID_SERV	NUMBER	Id del servicio	PK, NN, UA , FK(SERVICIOS)

Tabla 17. Relación alojamientos y servicios

Nombre	NUM_RESERV_ALOJ_SEM		
Descripción	Tabla que tiene la información de la cantidad de reservas en un alojamiento para cada semana		
Llave	ID_ALOJ, SEMANA, ANIO		
Atributo	Tipo	Descripción	Características/Restricciones
ID_ALOJ	NUMBER(19)	Id del alojamiento	PK, NN, SA , FK(ALOJAMIENTOS)
ID_OP	NUMBER(19)	Id del servicio	NN, SA
SEMANA	INTEGER	Semana del año (entre 1 y 52)	PK,NN, SA ,
ANIO	NUMBER(19)	Año	PK,NN, SA ,
NUM_RESERV	INTEGER	Número de reservas en el alojamiento en esa semana	NN, SA

Tabla 18. Tabla NUM_RESERV_ALOJ_SEM

3 Diseño Físico

3.1 Justificación de índices escogidos

Para la optimización de nuestras consultas, todos los índices, con una única excepción se implementaron sobre las *Foreign Keys* utilizadas para unir las distintas tablas requeridas en los requerimientos. Concretamente, se crearon los índices: aloj_id_op, en la tabla alojamientos, el índice es sobre la FK respectiva al operador del alojamiento; ofertas_id_al, en la tabla de ofertas de alojamientos, el índice es sobre la FK respectiva al alojamiento de la oferta; reserv_id_al_fech_creacion que es el índice que corresponde a la FK de una oferta de alojamiento en la tabla reservas; reserv_id_cliente que es la FK de un cliente en la tabla reservas; finalmente, el índice semana_anio sirve en la tabla NUM_RESERV_ALOJ_SEM.

Los índices definidos sobre *Foreign Keys* se pusieron en estas tablas para facilitar los múltiples Join's de alta selectividad necesarios en nuestras consultas. Por otro lado, el índice semana_anio, de la tabla NUM_RESERV_ALOJ_SEM se utilizó para facilitar las múltiples

consultas de altísima selectividad que se hacían sobre esta misma tabla en el requerimiento de consulta 12.

En este caso, todos los índices son primarios, de tipo árbol B+, esto corresponde a la forma como Oracle crea índices por defecto. Se decidió mantener la configuración por defecto de Oracle por dos sencillas razones, en primer lugar, por su mayor eficiencia de espacio frente a las demás alternativas. En segundo lugar, se debe tener en cuenta que la alternativa en Oracle es un bitmap, el cual es útil en casos en que muchos valores se agrupan bajo una misma llave, dado que este no es el caso en ninguno de nuestros índices, elegimos evitar esta implementación.

3.2 Índices Generados automáticamente por

Por defecto, Oracle crea un índice Árbol B+ para la PK de cada tabla y para las columnas con restricción ND.

A continuación, se muestran los índices generados automáticamente por Oracle en nuestra base de datos:

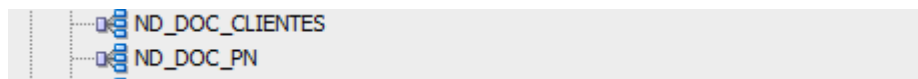


Figura 2. Índices generados por Oracle



Figura 3. Índices generados por Oracle

Dado que muchos de los joins que realizamos dependen de las PK, estos índices sí ayudan a mejorar la eficiencia de las consultas.

4 Documentación de los Requerimientos Funcionales y Análisis de eficiencia

4.1 Análisis de Eficiencia después de los índices y de las mejoras de eficiencia

RFC10:

Tiempo de Ejecución: 0.033 en promedio

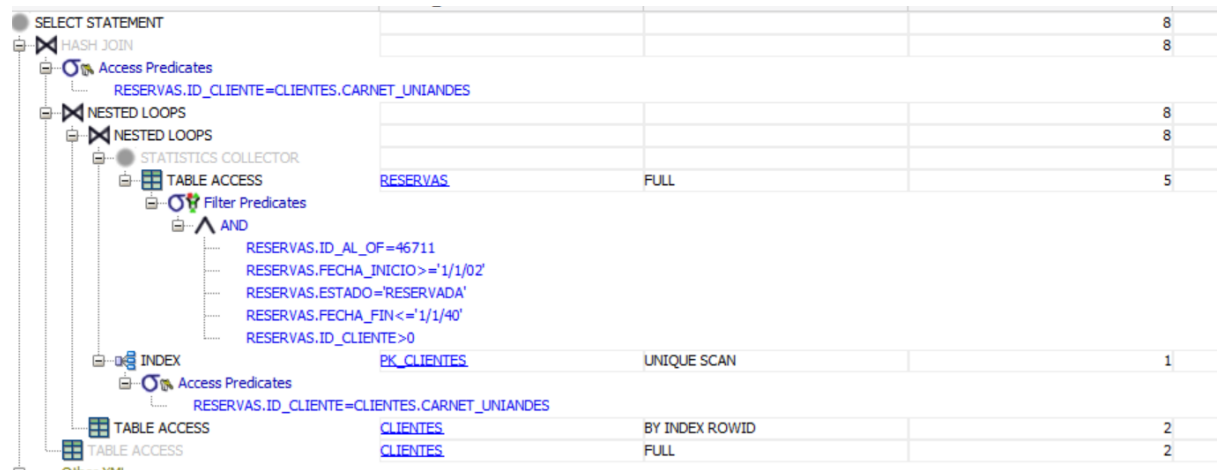


Figura 4. Plan de ejecución de RFC10

Análisis:

El resultado de esta consulta depende tanto de los parámetros de entrada como de la distribución de los datos creados en las tablas. En primer lugar, entre mayor sea el rango de fechas, existirán más usuarios que realizaron reservas en el alojamiento ya que entre mayor sea el rango cubrirá más reservas y por ende más usuarios. Por otro lado, también depende de la distribución de los datos en la base de datos ya que puede que haya muchas reservas para ese alojamiento en un rango de fechas dado, pero haya menos para otro alojamiento cualquiera en el mismo rango de fechas. Más aún, podría darse el caso de que existan muchas reservas en el rango de fecha pero que todas pertenezcan al mismo usuario lo cual daría un resultado más pequeño. Sin embargo, el plan de ejecución que se mostró anteriormente es independiente de los parámetros.

Plan de ejecución propuesto:

Parámetros: cota inferior del rango de fechas, cota superior del rango de fechas, id del alojamiento.

Escenario:

1. Juntar la tabla clientes con la tabla reservas (join) donde el carnet de uniandes de cada una de las tablas coincide.
2. De la tabla resultante tomar la subtabla compuesta por únicamente las tuplas donde ID_AL_OF es igual al id del alojamiento dado y el estado de la reserva es "RESERVADA".
3. Recorrer la sub-tabla, si en una tupla la fecha de inicio es menor o igual a la cota inferior dada y la fecha de fin es mayor o igual a la cota superior dada, añadir la tupla a la respuesta.

Comparación:

Oracle realiza el join con la metodología hash join, esto se debe a que la tabla clientes no cabe en memoria principal. Luego filtra esa tabla con todos los predicados que se colocaron en el where uno por uno, a diferencia de lo propuesto por nosotros que filtraba por ID_AL_OF y luego en una sola las fechas.

RFC11:

Tiempo de Ejecución: 0.7 en promedio

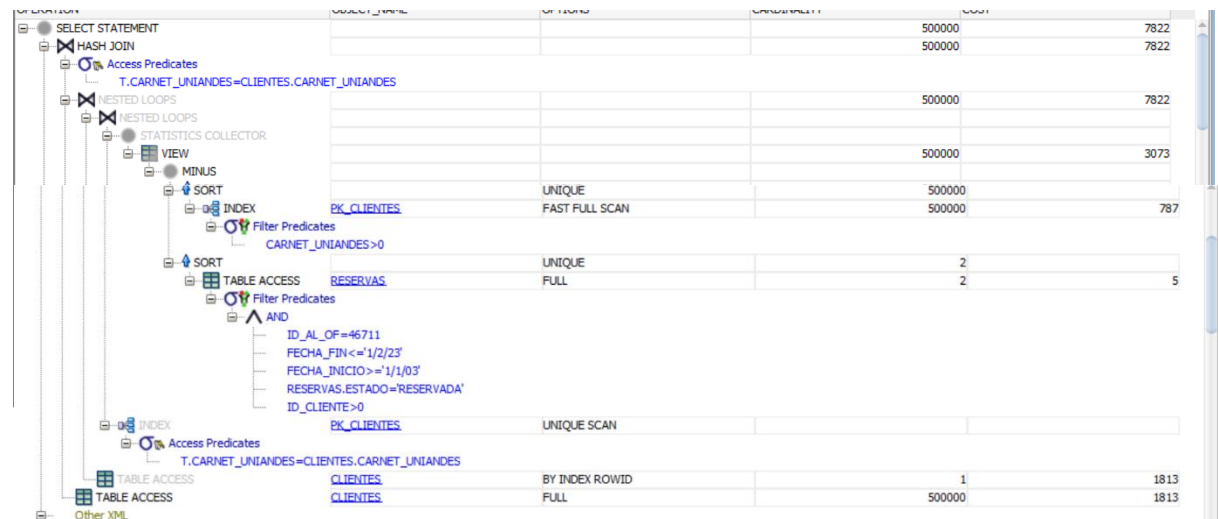


Figura 5. Plan de ejecución de RFC11

Análisis:

El resultado de este requerimiento depende fuertemente del rango de fechas que se dé por parámetro. Esto ya que entre mayor será el rango de fechas, existirán más clientes que hayan hecho reservas dentro del rango de fechas y por ende el resultado de la consulta será más pequeño. Además, también depende de la distribución de las reservas en la base de datos. Esto porque si hay pocas reservas en el rango de fechas dado, por más grande que sea el resultado será grande o incluso si hay varias, pero corresponden a el mismo cliente también se obtendrá un resultado grande. El plan de ejecución es independiente de los parámetros y la distribución de la base de datos.

Plan de ejecución propuesto:

Parámetros: cota inferior del rango de fechas, cota superior del rango de fechas, id del alojamiento.

Escenario:

1. Ejecutar el mismo plan que RFC1.
2. Seleccionar los ids de todos los clientes.
3. Cada cliente que se encuentre en la tabla generada en 2., pero no en 1. Añadirlo a la respuesta.

Comparación: Como puede observarse, el plan de ejecución elegido por Oracle es similar al propuesto, en primer lugar, Oracle selecciona todos los clientes y los ordena. A continuación, selecciona el id de los clientes que tienen una reserva que cumple con el predicado, es decir, cuyo estado es “RESERVADA” y del alojamiento y entre las fechas dadas por parámetro. Tras

esto, Oracle resta la primera tabla de la segunda y, para finalizar, le asigna a cada Id la información de su cliente correspondiente.

RFC12:

Tiempo de Ejecución: 0.13 en promedio



Figura 6. Plan de ejecución de RFC12

Análisis:

Esta consulta es dependiente del parámetro del año dado que para cada año los valores cambian. Sin embargo, el tamaño de la respuesta es igual sin importar el parámetro introducido ya que siempre el resultado tendrá solo 52 tuplas. En cuanto a la distribución de los datos de la base de datos, afectan a la respuesta ya que tienen un impacto en los valores de los resultados pero al igual que los parámetros, la distribución de la base de datos no afecta el tamaño del resultado.

Plan de ejecución:

Parámetros: año de consulta

1. Recorrer las 52 semanas del año.
2. Por cada semana buscar en la tabla las tuplas que corresponden a esa semana al año entrado por parámetro
3. Mirar cual es el alojamiento con más reservas y menos reservas de las tuplas seleccionadas
4. Sumar el número de reservas de cada operador
5. Mirar cual es el operador con más reservas y menos reservas de las tuplas seleccionadas

Comparación: En el plan de ejecución de Oracle se hacen 4 procesos diferentes uno para el alojamiento más ocupado, uno para el alojamiento menos ocupado, uno para el operador mas solicitado, y uno para el operador menos desocupado. En general sigue la misma estrategia, pero el plan de Oracle es más eficiente ya que aprovecha los índices implementados (SEMANA_ANIO) y así n tiene que buscar las tuplas sobre la tabla sino que accede ellas directamente.

RFC13

Tiempo de ejecución: 0.09 en promedio



Figura 6. Plan de ejecución de RFC12

Análisis:

Esta consulta es independiente de los parámetros por el simple hecho de que no hay parámetros. Por otro lado, si depende de la información que se encuentra en la base de datos ya que los clientes que cumplen con las condiciones de buenos clientes dependen estrictamente de la distribución de las reservas en la base de datos en cuanto a fechas, precios y características propias de las reservas. Por esta razón el tamaño del resultado de la consulta también depende de la distribución ya que el número de clientes depende de todas las cosas anteriormente mencionadas.

Plan de ejecución:

Parámetros: Ninguno.

1. Tomar todos los clientes registrados.
2. Por cada cliente contar el número de meses entre su creación y la fecha actual.
3. Juntar cada cliente con sus respectivas reservas.
4. Utilizando la tabla generada en 3, contar, para cada cliente, la cantidad de valores únicos del componente mes + año de la fecha de inicio de la reserva.
5. Usando las tablas generadas en 2 y 4, hacer un join, de modo que se hace el join únicamente si el coincide el carnet y si la cantidad de meses desde la creación del cliente coincide con lo contado en la consulta de 4.
6. Juntar cada cliente con sus respectivas reservas
7. Juntar cada cliente con sus reservas y cada reserva con su oferta.
8. De la tabla 8, tomar únicamente las ofertas cuyo precio es mayor a 150USD.
9. Crear una nueva tabla con los clientes presentes en 6 pero no en 8.
10. Juntar cada cliente con sus reservas y, si la reserva es de una habitación de hotel, juntarla con su respectivo alojamiento, si no, ignorar la tupla.
11. Por cada tupla de 10, por cada reserva, determinar si el tipo de la reserva es suite, si no lo es, descartar todo el cliente, si todas las reservas del cliente son en suites, añadir el cliente a la respuesta.
12. Unir las tablas de respuestas obtenidas en 5, 9 y 11.

Comparación: El proceso utilizado por Oracle es, de nuevo, muy similar al propuesto por nosotros. Oracle, primeramente, ejecuta cada una de las consultas a las que después se les hará union, se hace especial énfasis en el hecho de que, en estas consultas utilizar los índices. Una vez hecho esto, el sistema manejador de bases de datos junta todas las respuestas por medio de un Union.

5 Proceso de Carga de Datos

Para el proceso de carga de datos se desarrolló un programa de java que generaba información aleatoria pero que cumple con todas las reglas de negocio. El programa se encuentra adjunto con el nombre CIt3_C-09_m.devias_f.velasquez_creadorDeDatos. Este programa permitió poblar la mayoría de los datos, siendo una notoria excepción la tabla NUM_RESERV_ALOJ_SEM ya que esta tabla se implementó durante el proceso de optimización de las consultas. Para la inserción de datos de esta tabla en particular se implementó un nuevo servicio en nuestra aplicación que al ser llamado llena la tabla a partir de las reservas existentes en ese momento. Por otro lado, se modificó el servicio de agregar reserva y cancelar reserva para que cuando se invoquen estos servicios también se actualice NUM_RESERV_ALOJ_SEM y quede en estado coherente.

6 Análisis del proceso de optimización y el modelo de ejecución de consultas

En general, se puede decir que, al realizar las consultas, es preferible dejarle la tarea al sistema manejador de bases de datos. Esto se debe a varios factores. En primer lugar, debe considerarse que para que un programa manipule y filtre información, esta debe estar presente en la memoria principal de la máquina. Se puede dar el caso en el que algunas de las tablas son tan grandes que no caben en memoria principal. En casos así, el sistema manejador de bases de datos tiene la capacidad de realizar la consulta usando varios métodos de gran eficiencia. Aunque en teoría, es posible realizar esto mismo en el programa, la implementación de éstos mismos métodos no es para nada sencilla.

Adicionalmente, aún en los casos en que la memoria principal no es una restricción, se resalta que los distintos algoritmos de Joins utilizados por la base de datos son sumamente eficientes. Aunque se podrían implementar copias de estos algoritmos para hacer las consultas directamente en el programa, de nuevo, la alta complejidad de éstos hace que no sea práctico.

7 Resultados

7.1 Logrados

Se lograron cumplir todos los objetivos del proyecto, es decir, los requerimientos de consulta RF10, RF11, RF12 y RF13. Se destaca, en especial, que se logró que la ejecución de estas consultas se dé bajo el límite de tiempo estipulado.

7.2 No logrados

Consideramos que logramos todos los requerimientos satisfactoriamente.

8 Conclusiones

En conclusión, se pudo observar como el diseño físico de la base de datos puede influenciar enormemente en la eficiencia de las consultas. Se pudo observar claramente la relación existente entre costo espacial y temporal y como estas se pueden reorganizar al acomodo de los requerimientos no funcionales. También se puede concluir que la normalización de la base de datos no siempre va de la mano con la eficiencia, y que hay ocasiones en las que vale la pena perder la normalización para mejorar la eficiencia. Tanto los índices como la tabla incluida generan información redundante en la base de datos pero permitieron un incremento en velocidad muy notable. Sin embargo, al hacer estos cambios no hay que perder de vista la coherencia de la base de datos y hay que tomar acciones para garantizar que la información redundante quede coherente, de lo contrario la información deja de ser confiable.