Documentación entrega 2

Tomas Rodríguez - 202212868

Sergio Gonzalez - 202010910

Samuel Hernández - 202213772

Sistemas Transaccionales

Departamento de Ingeniería de Sistemas y Computación

Universidad de los Andes

Análisis 3
Requerimiento 1 (RCF1) 3
Requerimiento 2 (RCF2) 3
Requerimiento 3 (RCF3) 3
Requerimiento 4 (RCF4) 3
Requerimiento 5 (RCF5) 3
Requerimiento 6 (RCF6) 4
Requerimiento 7 (RCF7) 4
Requerimiento 8 (RCF8) 4
Requerimientos 9 y 10 (RCF9/RCF10) 4
Requerimiento 11 (RCF11) 4
Requerimiento 12 (RCF12) 4
Conclusiones 4
Diseño de la aplicación 4
Índices sugeridos 4
RFC1 - MOSTRAR EL DINERO RECOLECTADO POR SERVICIOS EN CADA HABITACIÓN EN EL ÚLTIMO AÑO CORRIDO: 5
RFC2 - MOSTRAR LOS 20 SERVICIOS MÁS POPULARES: 5
RFC3 - MOSTRAR EL ÍNDICE DE OCUPACIÓN DE CADA UNA DE LAS HABITACIONES DEL HOTEL: 5
RFC4 - MOSTRAR LOS SERVICIOS QUE CUMPLEN CON CIERTA CARACTERÍSTICA: 5
RFC5 - MOSTRAR EL CONSUMO EN HOTELANDES POR UN USUARIO DADO, EN UN RANGO DE FECHAS INDICADO: 5
RFC6 - ANALIZAR LA OPERACIÓN DE HOTELANDES 5
RFC7 - ENCONTRAR LOS BUENOS CLIENTES 6
RFC8 - ENCONTRAR LOS SERVICIOS QUE NO TIENEN MUCHA DEMANDA: 6
RFC9 - CONSULTAR CONSUMO EN HOTELANDES 6
RFC10 - RFC9v2: 6
RFC11 – Consultar funcionamiento: 6
RCF12 – Consultar clientes excelentes: 6
Indices creados por Oracle 6
Consultas 7
RFC1 - MOSTRAR EL DINERO RECOLECTADO POR SERVICIOS EN CADA HABITACIÓN EN EL ÚLTIMO AÑO CORRIDO: 7
RFC3 - MOSTRAR EL ÍNDICE DE OCUPACIÓN DE CADA UNA DE LAS HABITACIONES DEL HOTEL 8
RFC4 - MOSTRAR LOS SERVICIOS QUE CUMPLEN CON CIERTA CARACTERÍSTICA 9
RFC5 - MOSTRAR EL CONSUMO EN HOTELANDES POR UN USUARIO DADO, EN UN RANGO DE FECHAS INDICADO 10

RFC6 - ANALIZAR LA OPERACIÓN DE HOTELANDES 10
RFC7 - ENCONTRAR LOS BUENOS CLIENTES 11
RFC8 - ENCONTRAR LOS SERVICIOS QUE NO TIENEN MUCHA DEMANDA 13
RFC9 - CONSULTAR CONSUMO EN HOTELANDES 13
RFC10 - RFC9v2: 14
RFC11 – Consultar funcionamiento: 15
RCF12 – Consultar clientes excelentes: 15
Diseño y carga masiva de datos 17

Análisis

Requerimiento 1 (RCF1)

Mostrar el dinero recolectado por servicios en cada habitación en el último año corrido. Para esto, se debe agregar un atributo de fecha a la tabla de Consumo, lo que permitirá buscar en el último año.

Cambios: Agregar un atributo fecha a Consumo, para poder buscar en el último año recorrido

Requerimiento 2 (RCF2)

Mostrar los 20 servicios más populares. También es necesario agregar una fecha en la tabla de Consumo. Además, se debe considerar que es útil conocer el tipo de servicio sin tener que verificar si es gimnasio, piscina, tienda, etc. Se buscará en la tabla de Consumo para determinar cuántas veces aparece cada servicio.

Requerimiento 3 (RCF3)

Mostrar el índice de ocupación de cada una de las habitaciones del hotel. No se requieren cambios en este requerimiento.

Requerimiento 4 (RCF4)

Mostrar los servicios que cumplen con cierta característica. Se debe agregar una fecha a la tabla de Consumo.

Requerimiento 5 (RCF5)

Mostrar el consumo en HotelAndes por un usuario dado, en un rango de fechas indicado. Para esto, es necesario agregar un atributo de fecha a la tabla de Consumo para permitir la búsqueda en el rango de fechas.

Requerimiento 6 (RCF6)
Analizar la operación de HotelAndes. Se debe agregar un atributo de fecha a la tabla de Consumo para poder realizar análisis sobre fechas.

Requerimiento 7 (RCF7)
Encontrar los buenos clientes. Se necesita agregar un atributo de fecha a la tabla de Consumo para poder buscar en el último año.

Requerimiento 8 (RCF8)
Encontrar los servicios que no tienen mucha demanda. Se requiere agregar un atributo de fecha a la tabla de Consumo para buscar en el último año.

Requerimientos 9 y 10 (RCF9/RCF10)
Consultar consumo en HotelAndes. Es necesario agregar un atributo de fecha a la tabla de Consumo para permitir la búsqueda en el rango de fechas.

Requerimiento 11 (RCF11)
Consultar el funcionamiento. Agregar un atributo de fecha a la tabla de Consumo para permitir búsquedas en la semana.

Requerimiento 12 (RCF12)
Consultar los clientes excelentes. Se requiere agregar un atributo de fecha a la tabla de Consumo.

Conclusiones
Agregar el atributo de fecha a la tabla consumo, para poder filtrarlas. La mayoría de los requerimientos requieren esa fecha. Para cuando interese el tipo de consumo, como en el top de consumo, es más fácil si se tiene el atributo directamente en la tabla servicios.

Diseño de la aplicación

Índices sugeridos

RFC1 - MOSTRAR EL DINERO RECOLECTADO POR SERVICIOS EN CADA HABITACIÓN EN EL ÚLTIMO AÑO CORRIDO

Justificación de índice: Dado que esta consulta podría implicar el escaneo de grandes cantidades de datos, un índice secundario en la tabla "consumo" en la columna "fecha" sería útil para acelerar la búsqueda de datos dentro del último año.

Tipo de índice: Índice secundario (B-tree ya que es sobre rango) en la columna "fecha" de la tabla "consumo".

RFC2 - MOSTRAR LOS 20 SERVICIOS MÁS POPULARES

Justificación de índice: Un índice primario en la tabla "consumo" en la columna "servicio_id" es útil para contar la frecuencia de cada servicio consumido. El group by no se hace sobre otras columnas.

Tipo de índice: Índice secundario (B-tree) en la columna "servicio_id" de la tabla "consumo".

RFC3 - MOSTRAR EL ÍNDICE DE OCUPACIÓN DE CADA UNA DE LAS HABITACIONES DEL HOTEL

Justificación de índice: En la columna fechain de reserva, ya que hay que buscar en un rango de fechas (el ultimo año)

Tipo de índice: Índice secundario (B-tree) porque podría hacer repetición en las fechas.

RFC4 - MOSTRAR LOS SERVICIOS QUE CUMPLEN CON CIERTA CARACTERÍSTICA

Justificación de índice: La necesidad de un índice dependerá de las características específicas. Si las características incluyen precios o fechas, los índices en las columnas correspondientes podrían ser útiles. Sin embargo, ya que podrían darse varios datos repetidos en costo y fechas (en la vida real hay fechas con muchos servicios y costos iguales), se decidió no usar índices en estas consultas.

RFC5 - MOSTRAR EL CONSUMO EN HOTELANDES POR UN USUARIO DADO, EN UN RANGO DE FECHAS INDICADO

Justificación de índice: Un índice secundario en la tabla "reserva" en la columna "cliente_cliente_id" y en la tabla "consumo" en la columna "fecha" sería útil para acelerar la búsqueda de datos del usuario en el rango de fechas.

Tipo de índice: Índice secundario (B-tree ya que es en rangos) en la columna "cliente_cliente_id" de la tabla "reserva" y en la columna "fecha" de la tabla "consumo".

RFC6 - ANALIZAR LA OPERACIÓN DE HOTELANDES

Justificación de índice: Se requiere índice en las fechas (fecha de consumo y fechain de reserva) para optimizar las consultas en ambos casos (tanto en ocupación de habitaciones como en consumo)

Tipo de índice: Ambos b-tree, por la repetición.

RFC7 - ENCONTRAR LOS BUENOS CLIENTES

Justificación de índice: Un índice en la tabla "reserva" en la columna "cliente_cliente_id" haria que el producto cartesiano sea más eficiente.

Tipo de índice: Índice b-tree en la columna "cliente_cliente_id" de la tabla "reserva"

RFC8 - ENCONTRAR LOS SERVICIOS QUE NO TIENEN MUCHA DEMANDA

Justificación de índice: Un índice secundario en la tabla "consumo" en la columna "servicio_id" y en la columna "fecha" podría ayudar a acelerar la búsqueda de servicios con poca demanda.

Tipo de índice: Índice secundario (B-tree) en la columna "servicio_id" y "fecha" de la tabla "consumo".

RFC9 - CONSULTAR CONSUMO EN HOTELANDES

Justificación de índice: Para esta consulta, sería útil crear un índice secundario en la tabla "consumo" en la columna "cliente_cliente_id" y "servicio_id" para acelerar la búsqueda de clientes que consumieron un servicio específico en un rango de fechas.

Tipo de índice: Índice secundario (B-tree) en las columnas "cliente_cliente_id" y "servicio_id" de la tabla "consumo".

RFC10 - RFC9v2

justificación índice: Usar índice en “cliente_cliente_id”. En esta ocasión no hay un índice sobre servicio pues se buscarán los clientes que no consumieron.

Tipo de índice: Índice secundario (B-tree) en la columna "cliente_cliente_id"

RFC11 – Consultar funcionamiento

Justificación índice: Es útil crear índices en las fechas tanto de consumo como de reserva ya que la búsqueda es sobre rangos de fecha semanales.

Tipo índice: Índice secundario (B-tree) ya que es sobre rangos

RCF12 – Consultar clientes excelentes

Justificación índice: Las condiciones se verificarán desde reservasalon y en servicio. Para acceder más rápido a ellos desde clientes, se puede crear un índice en el id de reserva, que involucra ambos “caminos”.

Tipo índice: Índice secundario (B-tree)

Indices creados por Oracle

Consultas

RFC1 - MOSTRAR EL DINERO RECOLECTADO POR SERVICIOS EN CADA HABITACIÓN EN EL ÚLTIMO AÑO CORRIDO

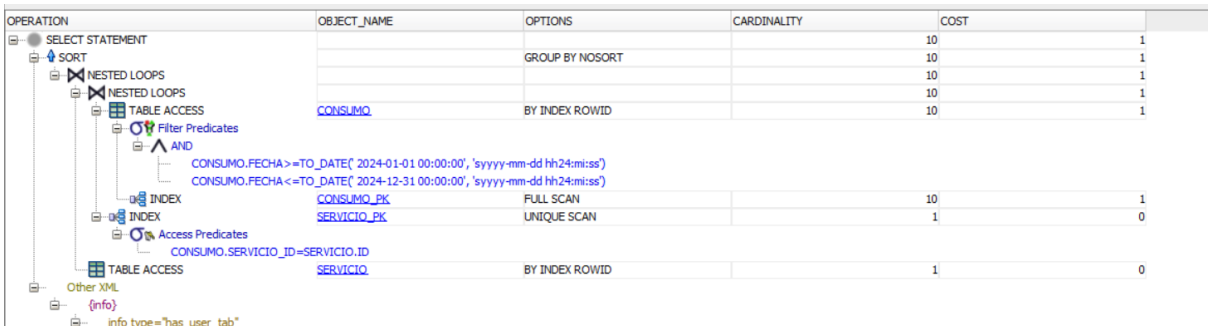
Resultados con fechas entre 1 de enero y 31 de diciembre

	CUENTA_HABITACION_ID	TOTALRECOLECTADO
1	52	14
2	155	191
3	474	186
4	536	120
5	709	163
6	734	173
7	812	173
8	928	20
9	956	149
10	969	129

Desde el 1 de junio hasta el 31 de diciembre

	CUENTA_HABITACION_ID	TOTALRECOLECTADO
1	52	14
2	155	191
3	474	186
4	928	20
5	956	149
6	969	129

Costos



Tiempo: 0,035

RFC2 - MOSTRAR LOS 20 SERVICIOS MÁS POPULARES

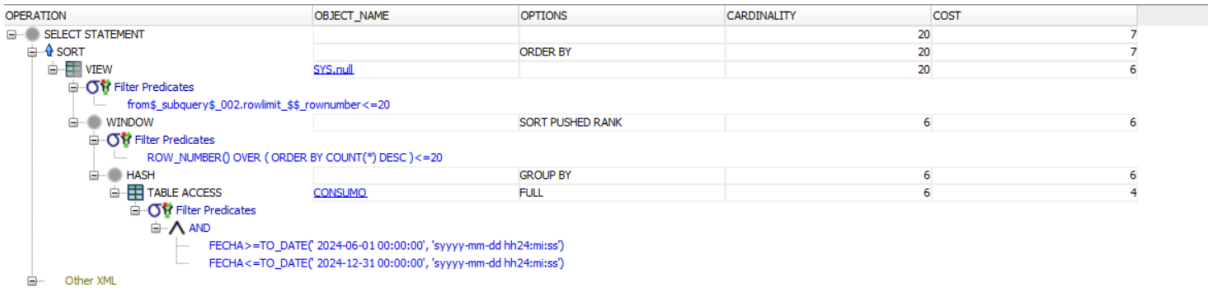
Resultados con fechas entre 1 de enero y 31 de diciembre

	SERVICIO_ID	NUMEROVECESCONSUMIDO
1	403	1
2	162	1
3	697	1
4	661	1
5	899	1
6	336	1
7	524	1
8	850	1
9	972	1
10	218	1

Desde el 1 de junio hasta el 31 de diciembre

	SERVICIO_ID	NUMEROVECESCONSUMIDO
1	403	1
2	661	1
3	899	1
4	336	1
5	850	1
6	218	1

Costos



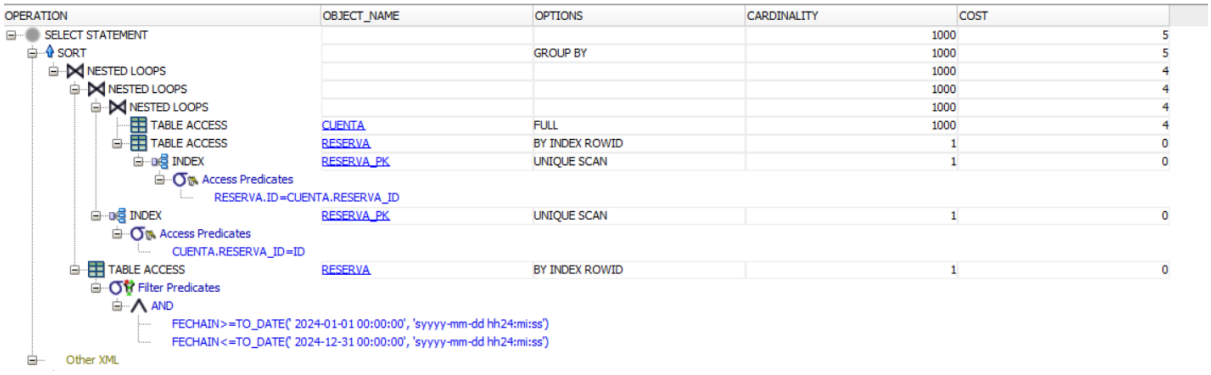
Tiempo: 0,036

RFC3 - MOSTRAR EL ÍNDICE DE OCUPACIÓN DE CADA UNA DE LAS HABITACIONES DEL HOTEL

Ocupación en el año 2024

	HABITACION_ID	PORCENTAJEOCUPACION
1	1	4,38
2	2	5,21
3	3	2,47
4	5	4,66
5	8	4,11
6	9	1,1
7	11	4,38
8	13	3,01
9	14	6,3
10	15	1,64
11	17	10,41
12	18	8,49
13	20	8,49

Costos



Tiempo: 0,116

RFC4 - MOSTRAR LOS SERVICIOS QUE CUMPLEN CON CIERTA CARACTERÍSTICA

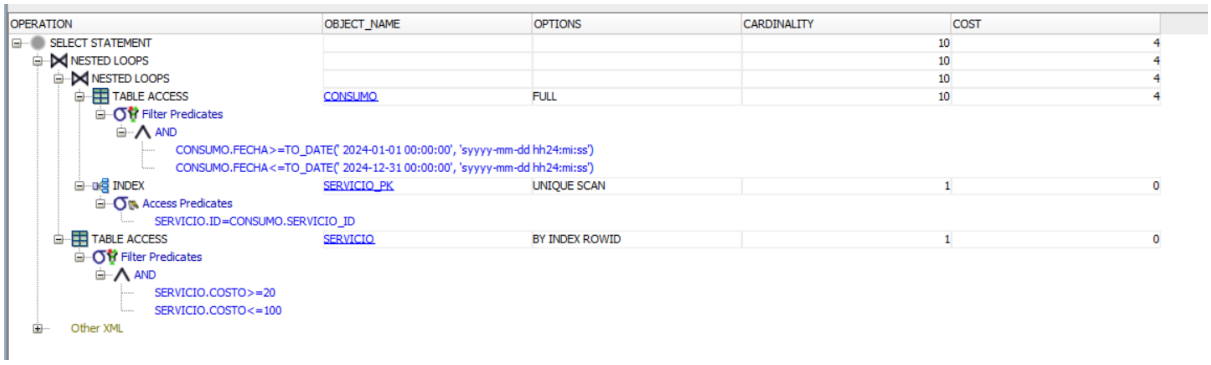
Servicios entre 0 y 100 de costo consumidos en el año 2024

	ID	COSTO	NOMBRE	CUENTA_HABITACION_ID	SERVICIO_ID	CUENTA_RESERVA_ID	FECHA
1	661	20	TIENDA	928	661	1	13/12/24
2	850	14	LOCAL	52	850	7	22/12/24

Servicios entre 20 y 100 de costo consumidos en el año 2024

	ID	COSTO	NOMBRE	CUENTA_HABITACION_ID	SERVICIO_ID	CUENTA_RESERVA_ID	FECHA
1	661	20	TIENDA	928	661	1	13/12/24

Costo



Tiempo: 0,033

RFC5 - MOSTRAR EL CONSUMO EN HOTELANDES POR UN USUARIO DADO, EN UN RANGO DE FECHAS INDICADO

Consumo de Antoinette (Cliente id 1) en el año 2024

	NOMBRE	NOMBRE_1	FECHA	COSTO
1	Antoinette	TIENDA	13/12/24	20

Consumo de Andrea (Cliente id 9) en el año 2024

	NOMBRE	SERVICIO	FECHA	COSTO
1	Eleanore	LOCAL	01/10/24	79
2	Eleanore	GYM	10/01/24	163

Costo

Tiempo: 0,043

RFC6 - ANALIZAR LA OPERACIÓN DE HOTELANDES

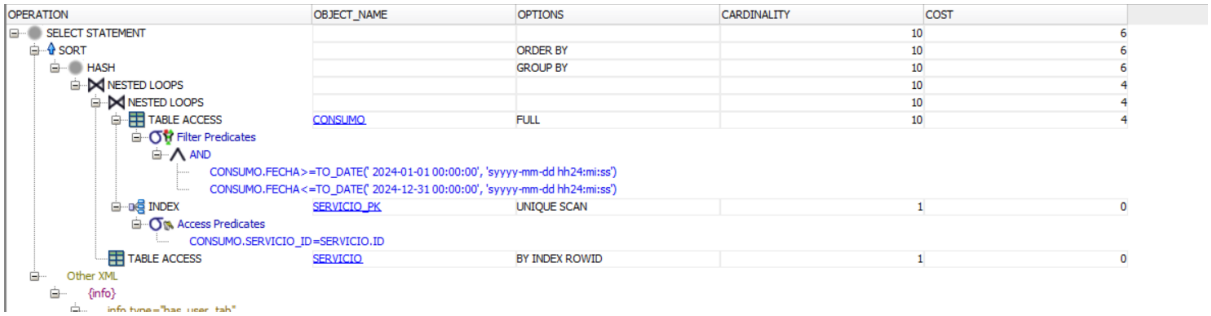
Fechas con mayores ingresos analizados en todo el año 2024

	FECHA	INGRESOS
1	27/06/24	191
2	19/10/24	186
3	20/05/24	173
4	20/01/24	173
5	10/01/24	163
6	16/12/24	149
7	31/07/24	129
8	02/05/24	120
9	13/12/24	20
10	22/12/24	14

Fechas con mayores ingresos analizados desde junio a diciembre del año 2024

	FECHA	INGRESOS
1	27/06/24	191
2	19/10/24	186
3	16/12/24	149
4	31/07/24	129
5	13/12/24	20
6	22/12/24	14

Costo



Tiempo: 0,396

RFC7 - ENCONTRAR LOS BUENOS CLIENTES

Buenos clientes en el periodo del año 2024

	CLIENTE_ID	TOTALGASTADO	TOTALNOCHES
1	2	0	19
2	7	0	14
3	8	0	14
4	11	0	14
5	13	0	14
6	19	0	20
7	22	0	14
8	23	0	17
9	25	0	20
10	26	0	16
11	27	0	15
12	33	0	17
13	38	0	16
14	41	0	19
15	43	0	18
16	49	0	20

Buenos clientes analizados desde junio a diciembre del año 2024

	CLIENTE_ID	TOTALGASTADO	TOTALNOCHES
1	2	0	19
2	7	0	14
3	8	0	14
4	11	0	14
5	13	0	14
6	22	0	14
7	23	0	17
8	25	0	20
9	26	0	16
10	27	0	15
11	38	0	16
12	43	0	18
13	49	0	20

Costos

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1000	1
FILTER				
Filter Predicates				
OR				
SUM(CUENTA.TOTAL)>15000000				
SUM(RESERVA.NUMNOCHES)>=14				
SORT		GROUP BY NOSORT	1000	1
NESTED LOOPS			1000	1
NESTED LOOPS			1000	1
TABLE ACCESS	RESERVA	BY INDEX ROWID	1000	1
Filter Predicates				
AND				
RESERVA.FECHAIN>=TO_DATE(' 2024-01-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss')				
RESERVA.FECHAIN<=TO_DATE(' 2024-12-31 00:00:00', 'syyyy-mm-dd hh24:mi:ss')				
INDEX	RESERVA_IDX	FULL SCAN	1000	1
INDEX	CUENTA_IDX	UNIQUE SCAN	1	0
Access Predicates				
RESERVA.ID=CUENTA.RESERVA_ID				
TABLE ACCESS	CUENTA	BY INDEX ROWID	1	0
Other XML				
{info}				
info type="has_user_tab"				

Tiempo: 0,114

RFC8 - ENCONTRAR LOS SERVICIOS QUE NO TIENEN MUCHA DEMANDA

Servicios menos demandados (menos de 3 veces semanales en promedio en el año 2024)

	SERVICIO_ID	VECESSOLICITADASPORSEMANA
1	403	0,019230769230769230769230769230769230769 2
2	162	0,0192307692307692307692307692307692307692
3	697	0,0192307692307692307692307692307692307692
4	661	0,0192307692307692307692307692307692307692
5	218	0,0192307692307692307692307692307692307692
6	336	0,0192307692307692307692307692307692307692
7	524	0,0192307692307692307692307692307692307692
8	850	0,0192307692307692307692307692307692307692
9	972	0,0192307692307692307692307692307692307692
10	899	0,0192307692307692307692307692307692307692

Servicios menos demandados (menos de 3 veces semanales en promedio en desde junio a diciembre de 2024)

	SERVICIO_ID	VECESSOLICITADASPORSEMANA
1	403	0,0192307692307692307692307692307692307692
2	661	0,0192307692307692307692307692307692307692
3	218	0,0192307692307692307692307692307692307692
4	336	0,0192307692307692307692307692307692307692
5	850	0,0192307692307692307692307692307692307692
6	899	0,0192307692307692307692307692307692307692

Costo

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			6	5
FILTER				
Filter Predicates				
COUNT(*)/52<3				
HASH		GROUP BY	6	5
TABLE ACCESS	CONSUMO	FULL	6	4
Filter Predicates				
AND				
FECHA>=TO_DATE(' 2024-06-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss')				
FECHA<=TO_DATE(' 2024-12-31 00:00:00', 'syyyy-mm-dd hh24:mi:ss')				
Other XML				

Tiempo: 0,119

RFC9 - CONSULTAR CONSUMO EN HOTELANDES

Producto id 700 consumido en el año 2024 ordenado el numero de veces que fue consumido

	CLIENTE_ID	EMAIL	NOMBRE	NUMEROVECESCONSUMIDO	ULTIMAFECHACONSUMO
1	1	aforbear0@elegantthemes.com	Antoinette	1	26/01/24
2	9	emollatt8@ovh.net	Eleanore	1	01/10/24

Producto id 700 consumido en el segundo semestre de 2024 ordenado el número de veces que fue consumido

	CLIENTE_ID	EMAIL	NOMBRE	NUMEROVECESCONSUMIDO	ULTIMAFECHACONSUMO
1	9	emollatt8@ovh.net	Eleanore	1	01/10/24

Costo

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	5
SORT		ORDER BY	1	5
HASH		GROUP BY	1	5
NESTED LOOPS			1	4
NESTED LOOPS			1	4
NESTED LOOPS			1	4
NESTED LOOPS			1	4
NESTED LOOPS			1	4
TABLE ACCESS	CONSUMO	FULL	1	4
Filter Predicates				
AND				
CONSUMO.FECHA>=TO_DATE(' 2024-06-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss')				
CONSUMO.FECHA<=TO_DATE(' 2024-12-31 00:00:00', 'syyyy-mm-dd hh24:mi:ss')				
CONSUMO.SERVICIO_ID=TO_NUMBER(:SERVICIOID)				
INDEX	CUENTA__IDX	UNIQUE SCAN	1	0
Access Predicates				
CUENTA.RESERVA_ID=CONSUMO.CUENTA_RESERVA_ID				
TABLE ACCESS	RESERVA	BY INDEX ROWID	1	0
INDEX	RESERVA_PK	UNIQUE SCAN	1	0
Access Predicates				
RESERVA.ID=CUENTA.RESERVA_ID				
TABLE ACCESS	CLIENTE	BY INDEX ROWID	1	0
INDEX	CLIENTE_PK	UNIQUE SCAN	1	0
Access Predicates				
CLIENTE.CLIENTE_ID=RESERVA.CLIENTE_CLIENTE_ID				
INDEX	USUARIO_PK	UNIQUE SCAN	1	0
Access Predicates				
CLIENTE.USUARIO_ID=USUARIO.ID				
TABLE ACCESS	USUARIO	BY INDEX ROWID	1	0
Other XML				

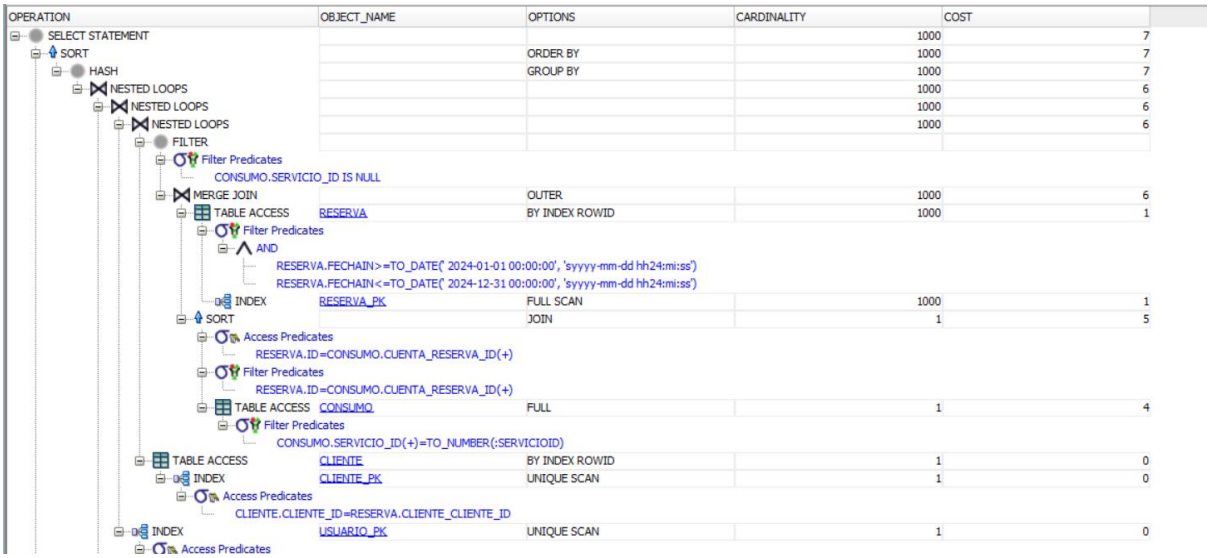
Tiempo: 0,103

RFC10 - RFC9v2

Clientes que no consumieron el servicio con id 700 en el año 2024-

	ID	NOMBRE
1	15	Thurstan
2	25	Johnathon
3	38	Sarina
4	51	El
5	56	Giraldo
6	59	Risa
7	78	Paolina
8	84	Keven

Costo



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1000	7
SORT		ORDER BY	1000	7
HASH		GROUP BY	1000	7
NESTED LOOPS			1000	6
NESTED LOOPS			1000	6
NESTED LOOPS			1000	6
FILTER				
Filter Predicates				
CONSUMO.SERVICIO_ID IS NULL				
MERGE JOIN		OUTER	1000	6
TABLE ACCESS	RESERVA	BY INDEX ROWID	1000	1
Filter Predicates				
AND				
RESERVA.FECHAIN>=TO_DATE(' 2024-01-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss')				
RESERVA.FECHAIN<=TO_DATE(' 2024-12-31 00:00:00', 'syyyy-mm-dd hh24:mi:ss')				
INDEX	RESERVA_PK	FULL SCAN	1000	1
SORT		JOIN	1	5
Access Predicates				
RESERVA.ID=CONSUMO.CUENTA_RESERVA_ID(+)				
Filter Predicates				
RESERVA.ID=CONSUMO.CUENTA_RESERVA_ID(+)				
TABLE ACCESS	CONSUMO	FULL	1	4
Filter Predicates				
CONSUMO.SERVICIO_ID(+)=TO_NUMBER(:SERVICIOID)				
TABLE ACCESS	CLIENTE	BY INDEX ROWID	1	0
INDEX	CLIENTE_PK	UNIQUE SCAN	1	0
Access Predicates				
CLIENTE.CLIENTE_ID=RESERVA.CLIENTE_CLIENTE_ID				
INDEX	USUARIO_PK	UNIQUE SCAN	1	0
Access Predicates				

Tiempo: 0,377

RFC11 – Consultar funcionamiento

RCF12 – Consultar clientes excelentes

Clientes que consumieron un servicio de mas de 50 dólares en su estancia o clientes que alquilaron salones en su estancia por 4 horas o más.

	USUARIOID	NOMBREUSUARIO	RAZON
1	1	Antoinette	Servicio costoso consumido
2	1	Antoinette	Servicio de salón
3	2	Tandi	Servicio costoso consumido
4	3	York	Servicio costoso consumido
5	4	Norine	Servicio costoso consumido
6	5	Marshal	Servicio costoso consumido
7	5	Marshal	Servicio de salón
8	6	Pebrook	Servicio costoso consumido
9	6	Pebrook	Servicio de salón
10	7	Dalenna	Servicio de salón
11	8	Leena	Servicio costoso consumido
12	8	Leena	Servicio de salón
13	9	Eleanore	Servicio costoso consumido
14	9	Eleanore	Servicio de salón
15	10	Andrea	Servicio costoso consumido
16	10	Andrea	Servicio de salón

Costo

Tiempo: 0,057

Diseño y carga masiva de datos

Para poblar la base de datos usamos la herramienta Mockaroo para generar datos aleatorios con cierto control (de manera procedural). Para otras tablas se decidió hacerlo de manera manual. Todas las sentencias están presentes en el proyecto y cada tabla tiene su archivo SQL. Las descripciones para cada tabla a están continuación:

- TipoUsuario: Se creo de manera manual para administrador, empleado y cliente
- Usuario: Se crearon 1.000 registros marcados como clientes con Mockaroo, y manualmente un administrador y 2 empleados.
- Cliente: Se crearon 1.000 registros usando los de usuario. Quedaron quedaran con el mismo id tanto en usuario como en cliente. En una situación normal esto no pasaría, pero cargarlo facilito el proceso y no mejora ni afecta otros tiempos.
- PlanConsumo: Se crearon manualmente, según la descripción del enunciado (todo incluido, larga estadía, etc.)
- Reserva: Se creo una reserva por cada cliente. Las fechas se acotaron entre 01/01/2024 y el 31/12/2024 y para tener coherencia con el planconsumo, se usó la siguiente sentencia:

```
if (numnoches) > 7 then generate('Number', min: 1, max: 3, decimals: 0)
  else generate('Number', min: 3, max: 4, decimals: 0) end
```

Si el número de noches es mayor a 7, puede ser cualquiera. Si es menor o igual, puede ser todo incluido o que no tiene plan (el plan con id=4 significa sin plan).

Para determinar el estado:

```
if fechain < date('01/03/2024','dd/mm/yyyy') then
  if date_diff('days', fechain, date('01/03/2024','dd/mm/yyyy')) < numnoches
    then 'activo'
    else 'cerrado' end
else 'creado' end
```

Se toma la fecha de 'hoy' como el primero de marzo. Con cálculos se determina si debe estar activa, cerrada o solo creada en ese punto.

- Cuenta: Se creo una cuenta por cada reserva y se uso Mockaroo para unirlo. La habitación se seleccionó de manera aleatoria entre los id de habitaciones que ya existían
- Salón y spa: Ambos se crearon manualmente ya que son pocos registros.
- Tipo habitación: Se crearon manualmente
- Habitación: Se crearon con Mockaroo, haciendo que el tipo habitación sea aleatorio entre los que existen.
- ReservaSalon: Se creo de manera manual, ya que una parte de su PK viene de una PK compuesta (de cuenta). Para que sean registros válidos, se revisa y registra uno por uno.
- Consumo: Al igual que con las reservas anteriores, se debe crear manual por su PK compuesto.
- Servicio: Se uso mockaroo. Se utilizaron pesos para distribuir el atributo tipo de servicio.

En este informe, se realizó un análisis de los requerimientos y consultas para el sistema transaccional relacionado con la gestión del Hotel de los Andes. Se identificaron y documentaron los requerimientos funcionales y se propusieron índices para optimizar el rendimiento de las consultas. Además, se ha proporcionado una visión general de los índices creados por Oracle en el entorno de la base de datos. El diseño y carga masiva de datos se ha llevado a cabo de manera eficiente, considerando la generación de datos aleatorios y la creación manual de registros necesarios para poblar las tablas.