Universidad de los Andes

FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS

ITERACIÓN 4: PROYECTO

Sergio Pardo Gutierrez
Juan Diego Yepes Parra

24 de mayo de 2022
Bogotá D.C.

Tabla de contenidos

1 Introducción

HOTELANDES

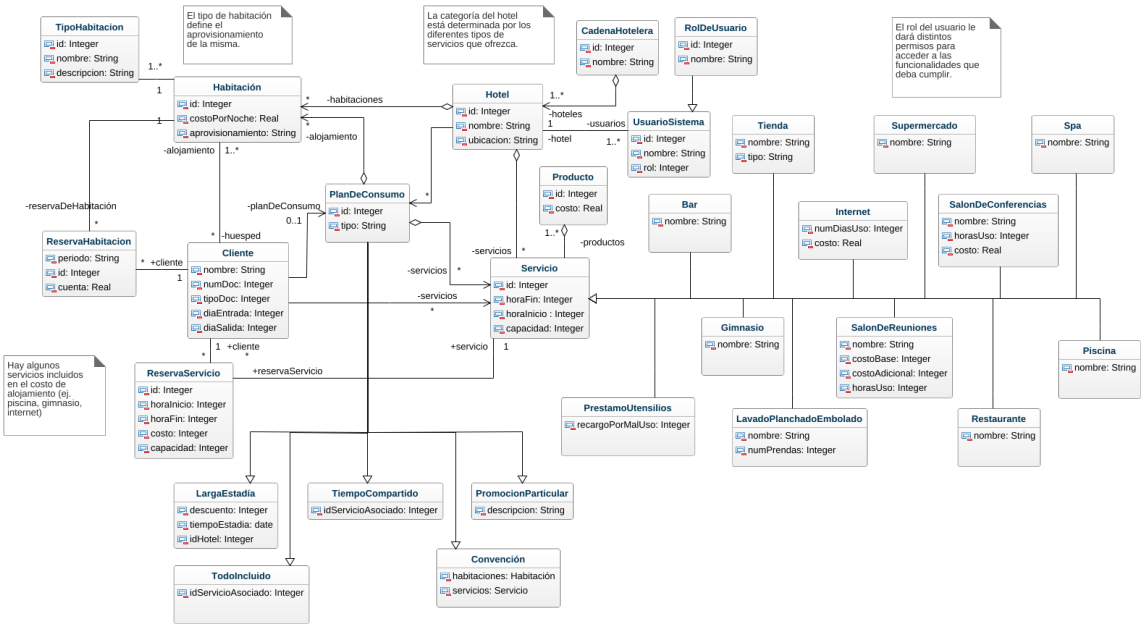
A-12

En el siguiente documento se pretende hacer un análisis de la cuarta iteración del proyecto HotelAndes; en el cual se van a detallar principalmente los cambios que estos nuevos requerimientos implican con respecto a la iteración anterior. En ese orden de ideas, primero se hablará del análisis conceptual del proyecto, ajustando el modelo conceptual y el relacional. Luego, se hablará de la construcción y el diseño del aplicativo con el fin de mostrar cuáles fueron los cambios que se hicieron.

2 Análisis

Teniendo en cuenta la retroalimentación recibida en la sustentación recibida en la sustentación de la iteración 3, este es el modelo relacional y conceptual planteado.

2.1 Modelo Conceptual



2.2 Modelo Relacional

Estas son las tablas de nuestro sistema:

- Hotel

id	nombre	ubicación
number	varchar2	varchar2
PK	UA	UA

- Cliente

nombre	numDoc	tipoDoc	diaEntrada	diaSalida
varchar2	number	varchar2	date	date
NN	PK, UA	PK, UA, CK in ('CC','TI','CE')	NC	NC

- TipoHabitacion

id	nombre	descripcion
number	varchar2	varchar2
PK	UA	UA

- Habitacion

id	costoPorNoche	aprovisionamiento	tipoHabitacion	mantenimiento
number	number	varchar2	number	varchar2
PK	UA	UA	FK$_{TipoHabitacion}$	CK in ('Y', 'N')

- RolesDeUsuario

id	nombre
number	varchar2
PK	UA

- UsuarioSistema

id	nombre	rol
number	varchar2	number
PK	UA	FK$_{RolesDeUsuario}$

- ReservaDeHabitacion

id	idHabitacion	tipoDocCliente	numDocCliente
number	number	varchar2	number
PK	FK$_{Habitacion}$	FK$_{Cliente}$, CK in ('CC', 'TI' ,'CE')	UA

diaEntrada	diaSalida	completada	cuenta
date	daate	varchar2	number
UA	UA	UA	UA

- Servicio

id	horaInicio	horaFin	capacidad	mantenimiento
number	timestamp	timestamp	number	varchar2
PK	UA, CK between 0 and 23	UA, CK between 0 and 23	UA	CK in ('Y', 'N')

tipoServicio
varchar2
FK $_{tipoServicio}$

- ReservaDeServicio

idReserva	idServicio	fechaInicio	fechaFin
number	number	date	date
PK, FK$_{ReservaHabitacion}$	FK$_{Servicio}$	UA	UA

- Piscina

idServicio	nombre
number	varchar 2
FK$_{Servicio}$, PK	UA

- Gimnasio

idServicio	nombre
number	varchar2
FK$_{Servicio}$, PK	UA

- Internet

idServicio	idReserva	numeroDiasUso	costo
number	number	number	number
$\texttt{FK}_{Servicio}$, PK	$\texttt{FK}_{Reserva}$, PK	UA	UA

- Bar

idServicio	nombre
number	varchar2
$\texttt{FK}_{Servicio}$, PK	UA

- Restaurante

idServicio	nombre	estilo
number	varchar2	varchar2
$\texttt{FK}_{Servicio}$, PK	UA	UA

- Supermercado

idServicio	nombre
number	varchar2
$\texttt{FK}_{Servicio}$, PK	UA

- Tienda

idServicio	nombre	tipo
number	varchar2	varchar2
$\texttt{FK}_{Servicio}$, PK	UA	UA

- Spa

idServicio	nombre
number	varchar2
$\texttt{FK}_{Servicio}$, PK	UA

- Lavado/planchado/embolado

idServicio	idReserva	tipoPrenda	numeroPrendas
number	number	varchar2	number
$\texttt{FK}_{Servicio}$, PK	$\texttt{FK}_{Reserva}$, PK	UA	UA

- PrestamoUtensilios

idServicio	idReserva	recargoPorMalUso
number	number	number
$\texttt{FK}_{Servicio}$, PK	$\texttt{FK}_{Reserva}$, PK	UA

- SalonReuniones

idServicio	idReserva	horasUso	costoBase	costoAdicionalPorEquipos
number	number	number	number	number
$\texttt{FK}_{Servicio}$, PK	$\texttt{FK}_{idReserva}$, PK	UA	UA	UA

- SalonConferencias

idServicio	idReserva	horasUso	costo
number	number	number	number
FK$_{Servicio}$, PK	FK$_{Reserva}$, PK	UA	UA

- Producto

id	costo
number	number
NN, ND, PK	UA

- ConsumoServicio

idFactura	idReserva	idServicio	idProducto	cantidad
number	number	number	number	number
PK	FK$_{ReservaHabitacion}$	FK$_{Servicio}$	PK, FK$_{Producto}$	UA

- PlanDeConsumo

id	tipo
number	varchar2
PK	UA

- LargaEstadía

idPlanDeConsumo	descuento	idHotel	tiempoEstadia
number	number	number	varchar2
PK	UA	FK$_{Hotel}$	UA

- TiempoCompartido

idPlanDeConsumo	idServicioAsociado
number	number
PK, FK$_{PlanDeConsumo}$	FK$_{Servicio}$

- TodoIncluido

idPlanDeConsumo	idServicioAsociado	cuentaHabitacion	costoFijoTotal
number	number	number	number
PK, FK$_{PlanDeConsumo}$	FK$_{Servicio}$	UA	UA

- ProductosTodoIncluido

idPlanDeConsumo	idProductoAsociado	descuento
number	number	number
PK, FK$_{PlanDeConsumo}$	FK$_{Producto}$	UA

- PromocionParticular

idPlanDeConsumo	descripcion
number	varchar2
PK, FK$_{PlanDeConsumo}$	UA

- Convencion

id	idPlanDeConsumo	numAsistentes	fechaInicio	fechaFin	cuenta	estado
number	number	number	date	date	number	varchar2
PK	, FK$_{PlanDeConsumo}$	UA	UA	UA		

- ConvencionServicio

idConvencion	idReservaServicio
number	number
PK, FK$_{Convencion}$	PK, FK$_{ReservaDeServicio}$

- ConvencionHabitacion

idConvencion	idReservaHabitacion
number	nuumber
PK, FK$_{Convencion}$	PK, FK$_{ReservaHabitacion}$

- TipoServicio

id	tipoServicio
number	varchar2
PK	NN, ND

3 Diseño y construcción de la aplicación

Para poder desarrollar los requerimientos que se detallan en el enunciado, realizamos las siguientes instrucciones e hicimos el siguiente análisis.

3.1 Impacto de introducción de nuevos requerimientos

Para el desarrollo de esta iteración se necesitaron agregar los siguientes requerimientos

3.1.1 Requerimientos funcionales de consulta

- RFC9 - CONSULTAR CONSUMO EN HOTELANDES

  La dificultad de implementar este requerimiento está en el dinamismo que hay que aplicarle a la consulta para poder añadir filtros y ordenar valores de acuerdo con lo que el usuario desee. A parte de esto, no fue necesario agregar nuevas tablas o atributos a las existentes.

- RFC10 - CONSULTAR CONSUMO EN HOTELANDES – RFC9-V2

  La dificultad de implementar este requerimiento fue bastante similar a la dificultad de aplicar el requerimiento 9, pues también era necesario crear una consulta que se pudiera dinamizar. De manera similar al requerimiento 9, no fue necesario crear nuevas tablas o atributos para que esta funcionara.

- RFC11 - CONSULTAR FUNCIONAMIENTO

  La dificultad de implementar este requerimiento está en 2 aspectos principalmente. Primero en la organización por semanas necesaria para el cumplir con el formato deseado. Esto implica hacer conversiones de fecha, agrupaciones y suma de algunos valores de los atributos. En segundo lugar, se encuentra la cantidad y variedad de información de información solicitada a distintas tablas que aparentemente no tienen nada en común. Esto desembocó en una consulta sql de cerca de 74 líneas de código con varios joins y uniones.

- RFC12 - CONSULTAR LOS BUENOS CLIENTES

  La dificultad de implementar este requerimiento fue principalmente la necesidad de separar esta consulta en varias subconsultas y juntarlas con el uso de joins. Esto, debido a que se buscaban clientes bajo distintos criterios.

3.1.2 Requerimientos no funcionales

- RNF6- EFICIENCIA EN LAS CONSULTAS

  La dificultad en este requerimiento está en que las consultas deben ser eficientes, deben cumplir con un límite de tiempo.

- RNF7- EFICIENCIA EN LA ACTUALIZACIÓN

  La aplicación debe garantizar eficiencia en la ejecución de los requerimientos de modificación solicitados.

- RNF8- ESQUEMA FÍSICO DE LA BASE DE DATOS

  El diseño físico de la base de datos debe reflejar un balance global de eficiencia de la aplicación, sin embargo, esto es algo de lo que se debe encargar Oracle por defecto.

3.2 Lógica de los nuevos requerimientos

3.2.1 Requerimientos funcionales de consulta

- RFC9 - CONSULTAR CONSUMO EN HOTELANDES

```
SELECT P.TIPODOC, P.NUMDOC, P.NOMBRE
FROM(
    SELECT C.TIPODOC, C.NUMDOC, C.NOMBRE, RH.DIAENTRADA, COUNT(CS.IDRESERVA)
        NUM_RESERVAS
    FROM HTA_CONSUMO_SERVICIO CS, HTA_CLIENTE C, HTA_RESERVA_HABITACION RH
    WHERE CS.IDRESERVA = RH.ID
        AND RH.NUMDOCCLIENTE = C.NUMDOC
        AND RH.TIPODOCCLIENTE = C.TIPODOC
        AND RH.DIAENTRADA > ('FECHAINICIAL')
        AND RH.DIASALIDA < ('FECHAFINAL')
        AND CS.IDSERVICIO = ('ID')
    GROUP BY C.TIPODOC, C.NUMDOC, C.NOMBRE, RH.DIAENTRADA) P

WHERE P.TIPODOC = ('TD')
WHERE P.NUMDOC = ('NUM')
WHERE P.NOMBRE = ('NOM')
WHERE P.DIAENTRADA = ('DIA')
WHERE NUM_RESERVAS = ('NUM')

AND P.TIPODOC = ('TD')
AND P.NUMDOC = ('NUM')
AND P.NOMBRE = ('NOM')
AND P.DIAENTRADA = ('DIA')
AND NUM_RESERVAS = ('NUM')

ORDER BY P.NUMDOC ASC
ORDER BY P.TIPODOC
ORDER BY P.NOMBRE
ORDER BY NUM_RESERVAS
ORDER BY P.DIAENTRADA

, P.NUMDOC ASC
, P.TIPODOC
, P.NOMBRE
, NUM_RESERVAS
, P.DIAENTRADA
```

La lógica de este requerimiento consiste en realizar una consulta con todos los datos sobre los cuáles se puede filtrar o realizar un ordenamiento (datos del cliente, día de la reserva y número de reservas), para después hacer una consulta únicamente con los datos necesarios y dejar

abierta la posibilidad de incluir filtros y ordenamientos. Esto es posible a través del uso de un contador que dé información acerca del número de filtro u ordenamiento. Si es el primer filtro u ordenamiento que se realiza, se coloca la palabra clave (WHERE o ORDER BY) y después de eso, solo se agregan los nuevos parámetros que el usuario desee.

- RFC10 - CONSULTAR CONSUMO EN HOTELANDES – RFC9-V2

```
SELECT DISTINCT P.TIPODOC, P.NUMDOC, P.NOMBRE
FROM (
    SELECT C.TIPODOC, C.NUMDOC, C.NOMBRE, RH.DIAENTRADA, RH.DIASALIDA
    FROM HTA_CLIENTE C, HTA_RESERVA_HABITACION RH
    WHERE RH.NUMDOCCLIENTE = C.NUMDOC
        AND RH.TIPODOCCLIENTE = C.TIPODOC
        AND (C.TIPODOC, C.NUMDOC, C.NOMBRE, RH.DIAENTRADA, RH.DIASALIDA) NOT
            IN (
        SELECT C.TIPODOC, C.NUMDOC, C.NOMBRE, RH.DIAENTRADA, RH.DIASALIDA
        FROM HTA_CONSUMO_SERVICIO CS, HTA_CLIENTE C, HTA_RESERVA_HABITACION
            RH
        WHERE CS.IDRESERVA = RH.ID
            AND RH.NUMDOCCLIENTE = C.NUMDOC
            AND RH.TIPODOCCLIENTE = C.TIPODOC
            AND RH.DIAENTRADA > ('DIAINICIO')
            AND RH.DIASALIDA < ('DIAFIN')
            AND CS.IDSERVICIO = 55
        GROUP BY C.TIPODOC, C.NUMDOC, C.NOMBRE, RH.DIAENTRADA,RH.DIASALIDA))
            P

WHERE P.TIPODOC = ('TD')
WHERE P.NUMDOC = ('NUM')
WHERE P.NOMBRE = ('NOM')

AND P.TIPODOC = ('TD')
AND P.NUMDOC = ('NUM')
AND P.NOMBRE = ('NOM')

ORDER BY P.NUMDOC ASC
ORDER BY P.TIPODOC
ORDER BY P.NOMBRE
ORDER BY P.DIAENTRADA

, P.NUMDOC ASC
, P.TIPODOC
, P.NOMBRE
, P.DIAENTRADA
```

La lógica detrás de este requerimiento es similar a la del requerimiento 9, se hace una consulta con todos los datos sobre los que posiblemente se puede realizar un filtro y se deja la posibilidad de agregar más según el usuario lo desee.

- RFC11 - CONSULTAR FUNCIONAMIENTO

```
SELECT E.SEMANA, ID_SER_MENOR, NUM_SER_MENOR, ID_SER_MAYOR, NUM_SER_MAYOR,
    ID_HAB_MENOR, NUM_RES_MENOR, ID_HAB_MAYOR, NUM_RES_MAYOR
FROM
     (SELECT A.SEMANA, ID_MENOR ID_SER_MENOR, A.NUM_RESERVAS NUM_SER_MENOR,
        ID_MAYOR ID_SER_MAYOR, B.NUM_RESERVAS NUM_SER_MAYOR
     FROM
         (SELECT SEMANA, ID ID_MENOR, NUM_RESERVAS
         FROM
         (SELECT ROW_NUMBER() OVER
         (PARTITION BY SEMANA  ORDER BY NUM_RESERVAS ASC) ROW_NUM, HC.*
         FROM(
             SELECT *
             FROM(
                 SELECT *
                 FROM(
                     SELECT to_number(to_char(to_date(to_char(cast(RS.
                         FECHAINICIO as date),'DD-MM-YYYY')),'WW')) SEMANA, S.
                         ID, COUNT(S.ID) NUM_RESERVAS
                     FROM  HTA_RESERVA_DE_SERVICIO RS, HTA_SERVICIO S
                     WHERE S.ID = RS.IDSERVICIO
                     GROUP BY S.ID, to_number(to_char(to_date(to_char(cast(RS.
                         FECHAINICIO as date),'DD-MM-YYYY')),'WW')))
                 UNION
                     (SELECT to_number(to_char(to_date(RH.DIAENTRADA),'WW'))
                         SEMANA, S.ID, COUNT(S.ID) NUM_RESERVAS
                     FROM  HTA_CONSUMO_SERVICIO CS, HTA_SERVICIO S,
                         HTA_RESERVA_HABITACION RH
                     WHERE S.ID = CS.IDSERVICIO
                     AND CS.IDRESERVA = RH.ID
                     GROUP BY S.ID, to_number(to_char(to_date(RH.DIAENTRADA),'
                         WW'))))
                 ORDER BY SEMANA, NUM_RESERVAS ASC)HC)
             WHERE ROW_NUM = 1)A
     FULL JOIN
     (SELECT SEMANA, ID ID_MAYOR, NUM_RESERVAS
     FROM
         (SELECT ROW_NUMBER() OVER
         (PARTITION BY SEMANA  ORDER BY NUM_RESERVAS DESC) ROW_NUM, HC.*
         FROM(
             SELECT *
         FROM(
             SELECT *
             FROM(
                 SELECT to_number(to_char(to_date(to_char(cast(RS.FECHAINICIO
                     as date),'DD-MM-YYYY')),'WW')) SEMANA, S.ID, COUNT(S.ID)
                     NUM_RESERVAS
                 FROM  HTA_RESERVA_DE_SERVICIO RS, HTA_SERVICIO S
                 WHERE S.ID = RS.IDSERVICIO
                 GROUP BY S.ID, to_number(to_char(to_date(to_char(cast(RS.
                     FECHAINICIO as date),'DD-MM-YYYY')),'WW')))
```

```
            UNION
                (SELECT to_number(to_char(to_date(RH.DIAENTRADA),'WW'))
                    SEMANA, S.ID, COUNT(S.ID) NUM_RESERVAS
                FROM  HTA_CONSUMO_SERVICIO CS, HTA_SERVICIO S,
                    HTA_RESERVA_HABITACION RH
                WHERE S.ID = CS.IDSERVICIO
                    AND CS.IDRESERVA = RH.ID
                GROUP BY S.ID, to_number(to_char(to_date(RH.DIAENTRADA),'WW')
                    )))
            ORDER BY SEMANA, NUM_RESERVAS DESC)HC)
        WHERE ROW_NUM = 1)B
    ON A.SEMANA = B.SEMANA)E
FULL JOIN
    (SELECT C.SEMANA, C.IDHABITACION ID_HAB_MENOR, C.NUM_RESERVAS
        NUM_RES_MENOR, D.IDHABITACION ID_HAB_MAYOR, D.NUM_RESERVAS
        NUM_RES_MAYOR
    FROM
        (SELECT SEMANA, IDHABITACION, NUM_RESERVAS
        FROM
        (SELECT ROW_NUMBER() OVER
        (PARTITION BY SEMANA  ORDER BY NUM_RESERVAS ASC) ROW_NUM, HC.*
        FROM
            (SELECT to_number(to_char(to_date(RH.DIAENTRADA),'WW')) SEMANA,
                IDHABITACION, COUNT(IDHABITACION) NUM_RESERVAS
            FROM HTA_RESERVA_HABITACION RH
            GROUP BY to_number(to_char(to_date(RH.DIAENTRADA),'WW')),
                IDHABITACION
            ORDER BY SEMANA, NUM_RESERVAS ASC) HC)
        WHERE ROW_NUM = 1) C
    FULL JOIN
        (SELECT SEMANA, IDHABITACION, NUM_RESERVAS
        FROM
        (SELECT ROW_NUMBER() OVER
        (PARTITION BY SEMANA  ORDER BY NUM_RESERVAS DESC) ROW_NUM, HC.*
        FROM
            (SELECT to_number(to_char(to_date(RH.DIAENTRADA),'WW')) SEMANA,
                IDHABITACION, COUNT(IDHABITACION) NUM_RESERVAS
            FROM HTA_RESERVA_HABITACION RH
            GROUP BY to_number(to_char(to_date(RH.DIAENTRADA),'WW')),
                IDHABITACION
            ORDER BY SEMANA, NUM_RESERVAS DESC) HC)
        WHERE ROW_NUM = 1) D
    ON D.SEMANA = C.SEMANA) F
ON E.SEMANA = F.SEMANA
ORDER BY SEMANA;
```

Esta consulta es la más extensa de todas. Primero se buscó hallar el número de reservas y consumos por semana, con su valor acumulado;

```
SELECT *
FROM(
```

```
        SELECT *
        FROM(
            SELECT to_number(to_char(to_date(to_char(cast(RS.FECHAINICIO as
                date),'DD-MM-YYYY')),'WW')) SEMANA, S.ID, COUNT(S.ID)
                NUM_RESERVAS
            FROM  HTA_RESERVA_DE_SERVICIO RS, HTA_SERVICIO S
            WHERE S.ID = RS.IDSERVICIO
            GROUP BY S.ID, to_number(to_char(to_date(to_char(cast(RS.
                FECHAINICIO as date),'DD-MM-YYYY')),'WW')))
            UNION
            (SELECT to_number(to_char(to_date(RH.DIAENTRADA),'WW')) SEMANA, S
                .ID, COUNT(S.ID) NUM_RESERVAS
            FROM  HTA_CONSUMO_SERVICIO CS, HTA_SERVICIO S,
                HTA_RESERVA_HABITACION RH
            WHERE S.ID = CS.IDSERVICIO
                AND CS.IDRESERVA = RH.ID
            GROUP BY S.ID, to_number(to_char(to_date(RH.DIAENTRADA),'WW'))))
    ORDER BY SEMANA, NUM_RESERVAS ASC)HC);
```

Como se puede ver se agrupa y se organiza en cada semana de menor a mayor el servicio reservado y su número de reservas. Se hace lo mismo para hallar el orden opuesto. Posteriormente se crean índices sobre el primer elemento de cada semana, dependiendo si se organizó de manera descendente o ascendente, se obtendrá el servicio con mayor o menor demanda esa semana.

```
(SELECT SEMANA, ID ID_MENOR, NUM_RESERVAS
FROM
    (SELECT ROW_NUMBER() OVER
    (PARTITION BY SEMANA  ORDER BY NUM_RESERVAS ASC) ROW_NUM, HC.*
    FROM(
    SELECT *
    FROM(
        SELECT *
        FROM(
            SELECT to_number(to_char(to_date(to_char(cast(RS.FECHAINICIO as
                date),'DD-MM-YYYY')),'WW')) SEMANA, S.ID, COUNT(S.ID)
                NUM_RESERVAS
            FROM  HTA_RESERVA_DE_SERVICIO RS, HTA_SERVICIO S
            WHERE S.ID = RS.IDSERVICIO
            GROUP BY S.ID, to_number(to_char(to_date(to_char(cast(RS.
                FECHAINICIO as date),'DD-MM-YYYY')),'WW')))
            UNION
            (SELECT to_number(to_char(to_date(RH.DIAENTRADA),'WW')) SEMANA, S
                .ID, COUNT(S.ID) NUM_RESERVAS
            FROM  HTA_CONSUMO_SERVICIO CS, HTA_SERVICIO S,
                HTA_RESERVA_HABITACION RH
            WHERE S.ID = CS.IDSERVICIO
                AND CS.IDRESERVA = RH.ID
            GROUP BY S.ID, to_number(to_char(to_date(RH.DIAENTRADA),'WW'))))
    ORDER BY SEMANA, NUM_RESERVAS ASC)HC)
WHERE ROW_NUM = 1)A
```

Finalmente, se hace un join entre ambos datos y se deja en un mismo query el servicio mayor y menor por semana.

Se reliza un proceso similar con las habitaciones:

```
SELECT SEMANA, IDHABITACION, NUM_RESERVAS
FROM
    (SELECT ROW_NUMBER() OVER
        (PARTITION BY SEMANA  ORDER BY NUM_RESERVAS ASC) ROW_NUM, HC.*
        FROM
            (SELECT to_number(to_char(to_date(RH.DIAENTRADA),'WW')) SEMANA,
                IDHABITACION, COUNT(IDHABITACION) NUM_RESERVAS
            FROM HTA_RESERVA_HABITACION RH
            GROUP BY to_number(to_char(to_date(RH.DIAENTRADA),'WW')),
                IDHABITACION
            ORDER BY SEMANA, NUM_RESERVAS ASC) HC)
WHERE ROW_NUM = 1
```

Y finalmente se junta todo esto para obtener la consulta mostrada al comienzo. El resultado con la base de datos totalmente cargada es el siguiente:

	SEMANA	ID_SER_MENOR	NUM_SER_MENOR	ID_SER_MAYOR	NUM_SER_MAYOR	ID_HAB_MENOR	NUM_RES_MENOR	ID_HAB_MAYOR	NUM_RES_MAYOR
1	1	53	117	43	600	178504	250	178504	250
2	2	44	89	46	800	124163	250	124163	250
3	3	52	91	49	400	199920	250	199920	250
4	4	63	109	43	500	172744	250	172744	250
5	5	37	108	49	400	160079	250	160079	250
6	6	57	107	40	650	84817	250	84817	250
7	7	62	96	40	450	411	250	411	250
8	8	41	81	40	500	19448	250	19448	250
9	9	49	100	37	300	73315	250	73315	250
10	10	48	119	40	450	22495	250	22495	250
11	11	60	110	43	500	192160	250	192160	250
12	12	29	80	43	450	44812	250	44812	250
13	13	55	100	49	700	101882	250	101882	250
14	14	46	50	43	500	54581	250	54581	250
15	15	58	91	37	450	164616	250	164616	250
16	16	37	84	49	800	464	250	464	250
17	17	28	93	49	600	53179	250	53179	250
18	18	28	100	40	500	190406	250	190406	250
19	19	28	89	40	450	95827	250	95827	250
20	20	51	125	43	400	156595	250	156595	250
21	21	46	107	37	600	51912	250	51912	250
22	22	47	87	49	350	185375	250	185375	250
23	23	62	126	37	500	114317	250	114317	250
24	24	32	118	40	700	18488	250	18488	250
25	25	29	107	43	450	115288	250	115288	250
26	26	38	72	40	600	97270	250	97270	250
27	27	46	100	55	500	77732	250	77732	250
28	28	43	153	49	600	41394	250	41394	250
29	29	30	129	40	400	80462	250	80462	250
30	30	45	107	46	450	169644	250	169644	250
31	31	42	104	37	500	15112	250	15112	250
32	32	36	88	49	300	119289	250	119289	250
33	33	46	50	49	300	153786	250	153786	250
34	34	46	100	43	550	109466	250	109466	250
35	35	40	100	37	550	143225	250	143225	250
36	36	46	50	40	400	124094	250	124094	250
37	37	64	113	55	550	157807	250	157807	250

ID_SER_MAYOR hace referencia al id del servicio con más demanda y NUM_SER_MAYOR El número de reservas que tuvo. Lo mismo sucede con el servicio menor y las habitaciones.

- RFC12 - CONSULTAR LOS BUENOS CLIENTES

  La lógica de este requerimiento se encuentra en hacer una consulta sobre los clientes que cubren los criterios que se piden por parámetro.

```
select C.TIPODOC, C.NUMDOC, C.NOMBRE
from
(select p.id
from hta_servicio p, HTA_SERVICIO_INTERNET i
where (p.tiposervicio = 3)
    and (p.id = i.idServicio)
    and (i.costo >= 300)
union all
select p.id
```

```
from hta_servicio p, HTA_SALON_REUNIONES sr
where (p.tiposervicio = 11)
    and (p.id = sr.idServicio)
    and (sr.costobase+sr.costo_adicional >= 300)
union all
select p.id
from hta_servicio p, HTA_SALON_CONFERENCIAS sc
where (p.tiposervicio = 12)
    and (p.id = sc.idServicio)
    and (sc.costo >= 300)) a,
HTA_CLIENTE C,
HTA_CONSUMO_SERVICIO co,
HTA_RESERVA_HABITACION RH
WHERE RH.NUMDOCCLIENTE = C.NUMDOC
AND RH.TIPODOCCLIENTE = C.TIPODOC
and rh.id = co.idreserva
and co.idservicio = a.id;
```

Por ejemplo, en la consulta anterior se buscan los clientes que gastan más de 300 mil pesos en servicios. (Solo en los servicios que tienen un costo)

3.2.2 Requerimientos no funcionales

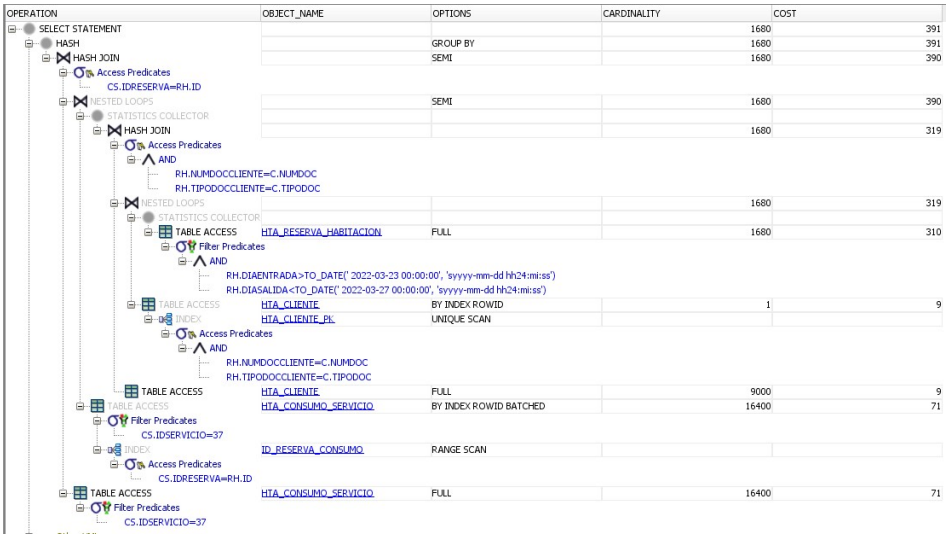
- RNF6- EFICIENCIA EN LAS CONSULTAS, RNF7 EFICIENCIA EN LA ACTUALIZACIÓN

  Las tablas utilizadas para las consultas fueron las que más recibieron datos al momento de poblar el sistema con un millón de registros. Se crearon varios índices para hacer más eficientes las consultas. En promedio, nuestras consultas se demoran 0.05 segundos, lo que demuestra gran eficiencia. En particular, se agregaron:

  - `id_tipo_hab`
  - `mantenimiento_hab`
  - `rol_usuario`
  - `id_habitacion`
  - `reser_completada`
  - `tipo_doc`
  - `tipo_serv`
  - `id_reserva_consumo`
  - `id_servicio`
  - `id_reserva_reserva`
  - `id_servicio`

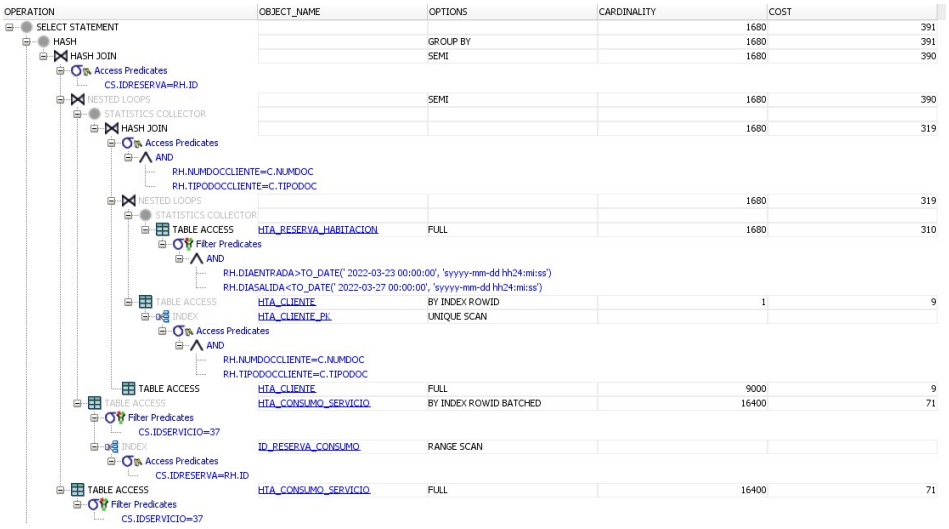
  Estos índices se pueden consultar en el archivo `data\indices.sql`

  Aquí se puede ver reflejada la eficiencia en los siguientes planes de ejecución de las consultas:



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1680	391
HASH		GROUP BY	1680	391
HASH JOIN		SEMI	1680	390
Access Predicates				
CS.IDRESERVA=RH.ID				
NESTED LOOPS		SEMI	1680	390
STATISTICS COLLECTOR				
HASH JOIN			1680	319
Access Predicates				
AND				
RH.NUMDOCCLIENTE=C.NUMDOC				
RH.TIPODOCCLIENTE=C.TIPODOC				
NESTED LOOPS			1680	319
STATISTICS COLLECTOR				
TABLE ACCESS	HTA_RESERVA_HABITACION	FULL	1680	310
Filter Predicates				
AND				
RH.DIAENTRADA>TO_DATE(' 2022-03-23 00:00:00', 'syyyy-mm-dd hh24:mi:ss')				
RH.DIASALIDA<TO_DATE(' 2022-03-27 00:00:00', 'syyyy-mm-dd hh24:mi:ss')				
TABLE ACCESS	HTA_CLIENTE	BY INDEX ROWID	1	9
INDEX	HTA_CLIENTE_PK	UNIQUE SCAN		
Access Predicates				
AND				
RH.NUMDOCCLIENTE=C.NUMDOC				
RH.TIPODOCCLIENTE=C.TIPODOC				
TABLE ACCESS	HTA_CLIENTE	FULL	9000	9
TABLE ACCESS	HTA_CONSUMO_SERVICIO	BY INDEX ROWID BATCHED	16400	71
Filter Predicates				
CS.IDSERVICIO=37				
INDEX	ID_RESERVA_CONSUMO	RANGE SCAN		
Access Predicates				
CS.IDRESERVA=RH.ID				
TABLE ACCESS	HTA_CONSUMO_SERVICIO	FULL	16400	71
Filter Predicates				
CS.IDSERVICIO=37				

Como se puede observar, para la consulta 9 se utilizan 2 de los índices insertados en la base de datos, mejorando la eficiencia de la misma.



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1680	391
HASH		GROUP BY	1680	391
HASH JOIN		SEMI	1680	390
Access Predicates				
CS.IDRESERVA=RH.ID				
NESTED LOOPS		SEMI	1680	390
STATISTICS COLLECTOR				
HASH JOIN			1680	319
Access Predicates				
AND				
RH.NUMDOCCLIENTE=C.NUMDOC				
RH.TIPODOCCLIENTE=C.TIPODOC				
NESTED LOOPS			1680	319
STATISTICS COLLECTOR				
TABLE ACCESS	HTA_RESERVA_HABITACION	FULL	1680	310
Filter Predicates				
AND				
RH.DIAENTRADA>TO_DATE(' 2022-03-23 00:00:00', 'syyyy-mm-dd hh24:mi:ss')				
RH.DIASALIDA<TO_DATE(' 2022-03-27 00:00:00', 'syyyy-mm-dd hh24:mi:ss')				
TABLE ACCESS	HTA_CLIENTE	BY INDEX ROWID	1	9
INDEX	HTA_CLIENTE_PK	UNIQUE SCAN		
Access Predicates				
AND				
RH.NUMDOCCLIENTE=C.NUMDOC				
RH.TIPODOCCLIENTE=C.TIPODOC				
TABLE ACCESS	HTA_CLIENTE	FULL	9000	9
TABLE ACCESS	HTA_CONSUMO_SERVICIO	BY INDEX ROWID BATCHED	16400	71
Filter Predicates				
CS.IDSERVICIO=37				
INDEX	ID_RESERVA_CONSUMO	RANGE SCAN		
Access Predicates				
CS.IDRESERVA=RH.ID				
TABLE ACCESS	HTA_CONSUMO_SERVICIO	FULL	16400	71
Filter Predicates				
CS.IDSERVICIO=37				

Lo mismo sucede con el requerimiento de consulta número 10.

- RNF8- ESQUEMA FÍSICO DE LA BASE DE DATOS

  Como se dijo anteriormente, este requerimiento lo provee Oracle por defecto.

4 Referencias

Oracle (2022) *Database SQL Reference*. Desde https://docs.oracle.com/cd/B19306_01/server.102/b14200/toc.htm

Oracle (2003) *Regexp_Like*. Desde https://docs.oracle.com/cd/B12037_01/server.101/b10759/conditions018.htm

Oracle (S.A) *4 Using Regular Expressions y Oracle Database*. Desde https://docs.oracle.com/cd/B19306_01/B1425101/adfnsregexp.htm

Hebbrecht, J. (2022). *Drop all tables in Oracle DB (scheme)* | JOCHEN HEBBRECHT. Recuperado de http://www.jochenhebbrecht.be/site/2010-05-10/database/drop-all-tables-in-oracle-db-scheme