Universidad de los Andes

FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS

ITERACIÓN 3: PROYECTO

Sergio Pardo Gutierrez
Juan Diego Yepes Parra

5 de mayo de 2022
Bogotá D.C.

Tabla de contenidos

1 Introducción

HOTELANDES

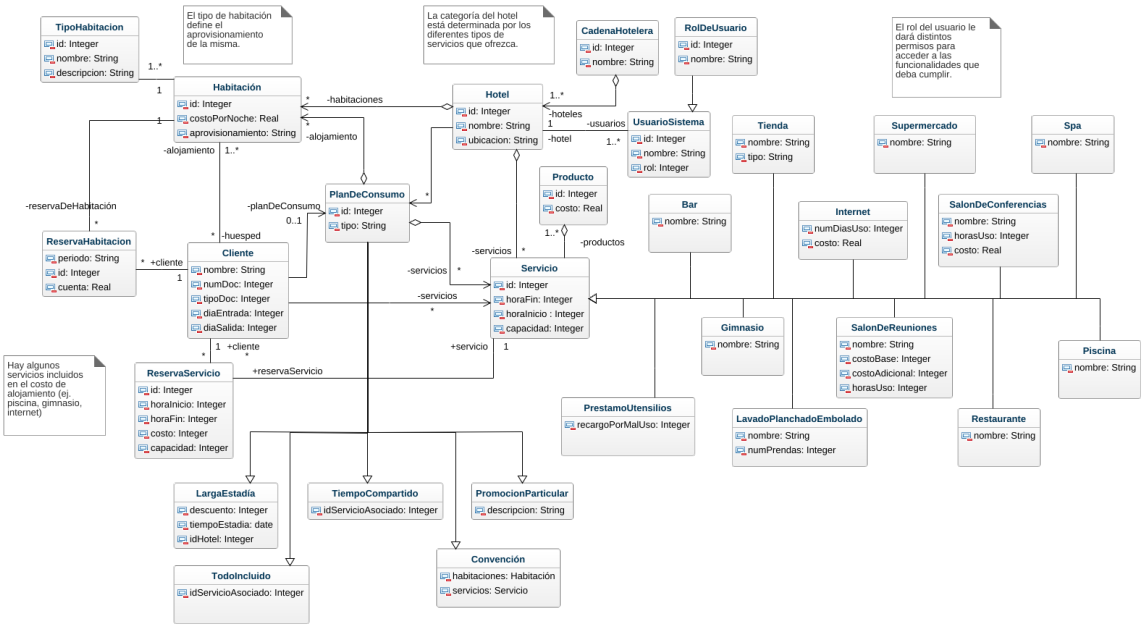
A-12

En el siguiente documento se pretende hacer un análisis de la tercera 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 2 y los cambios necesarios para esta iteración, estos son los cambios hechos en el proceso de modelado

2.1 Modelo Conceptual



Lo que hicimos diferente con respecto a la iteración pasada es que agregamos el plan de consumo de convención y arreglamos algunos atributos que no tenían sentido.

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
$\texttt{FK}_{Servicio}$, 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
FK$_{Servicio}$, PK	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, $\mathrm{FK}_{PlanDeConsumo}$	UA

- **Convencion**

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

- **ConvencionServicio**

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

- **ConvencionHabitacion**

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

- **TipoServicio**

id	tipoServicio
number	varchar2
PK	NN, ND

En comparación con las tablas anteriores, hemos adicionado las tablas en negrilla; esto para satisfacer los nuevos requerimientos de la iteración. De igual modo, también agregamos algunos atributos, como para las tablas Habitación y Servicio, ya que necesitábamos saber si estaban en mantenimiento o no para poder hacer las reservas.

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

- **RESERVAR ALOJAMIENTO Y SERVICIOS PARA UNA CONVENCIÓN**

  Para este requerimiento era necesario tener varias cosas. Primero, era necesario poder construir una convención de acuerdo con los parámetros dados, luego poder asociar todas las reservas a esa convención y finalmente poder hacer que las cuentas de cobro se unificaran para esa convención.

- **CANCELAR RESERVAS ASOCIADAS A UNA CONVENCIÓN**

  Para este requerimiento era necesario poder ver las reservas de una convención que debieran ser canceladas; y eliminarlas de la base de datos.

- **REGISTRAR EL FIN DE UNA CONVENCIÓN**

  Para este requerimiento era necesario tener algún indicativo de si la convención estaba iniciada o cerrada, tal y como una reserva o como un plan de consumo.

- **REGISTRAR LA ENTRADA A MANTENIMIENTO DE ALOJAMIENTOS O SERVICIOS DEL HOTEL**

  Para este requerimiento era necesario tener algún indicativo de si el servicio estaba en uso y si estaba en mantenimiento, para con el fin de desplazar los consumos de dicho cliente a otra habitación.

- **REGISTRAR EL FIN DEL MANTENIMIENTO DE ALOJAMIENTOS O SERVICIOS DEL HOTEL**

  Para este requerimiento es necesario cumplir con el requerimiento anterior.

3.1.2 Requerimientos funcionales de consulta

- **ANALIZAR LA OPERACIÓN DE HOTELANDES**

  Esta consulta fue la que impuso un reto más grande para ser completada, pues además de la dificultad de la consulta, fue necesario cambiar una de las entidades principales de nuestro modelo relacional, el servicio. Para la iteración anterior no se tenía un atributo tipo de servicio ni una tabla dedicada para asociar los tipos de servicio con los propuestos en el enunciado. Se creo la tabla tipoServicio y se le agregó un atributo a servicio que le asocia un tipo de servicio al momento de ser agregado.

- **ENCONTRAR LOS BUENOS CLIENTES**

  La dificultad de esta consulta yacía en los filtros que era necesario aplicar para obtener la información deseada, puesto que no fue necesario hacer ajustes a nuestro modelo. Para hallar el tiempo de estadía total del cliente en el último año se utilizaron las fechas presentes en nuestra tabla reservaHabitación y para el consumo total del cliente se utilizó el atributo cuenta asociado a una reserva en este misma tabla, donde se guarda el consumo de cada cliente a lo largo de su estadía.

- **ENCONTRAR LOS SERVICIOS QUE NO TIENEN MUCHA DEMANDA**

  De la misma manera, esta consulta no requirió cambios en nuestro modelo puesto que ya se contaba con la tabla reservaServicio que contiene una fecha de reserva. De esta forma, se tiene toda la información necesaria para cumplir esta consulta. Por este motivo, la dificultad de esta consulta yace también en traducir estos filtros a lenguaje sql.

3.1.3 Requerimientos no funcionales

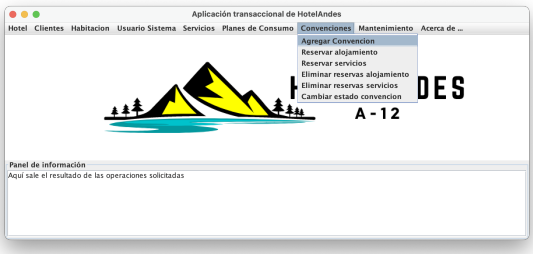
Para esta iteración el requerimiento no funcional que se debía asegurar era la **transaccionalidad**.

3.2 Lógica de los nuevos requerimientos

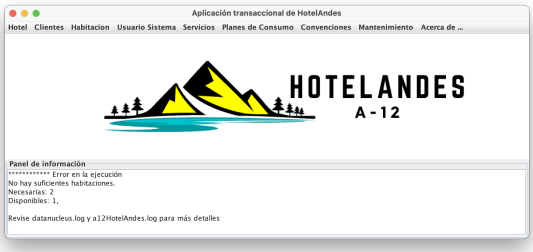
3.2.1 Requerimientos funcionales

- **RESERVAR ALOJAMIENTO Y SERVICIOS PARA UNA CONVENCIÓN**

  Para este requerimiento lo que se hacen son varias operaciones transaccionales secuenciales. En primer lugar, un usuario de tipo administrativo debe registrar al cliente de tipo organizador de convención. Esto lo hace registrando a un cliente normalmente. Luego, el usuario debe registrar el inicio de una convención; esto lo hace mediante el siguiente orden lógico en el aplicativo:

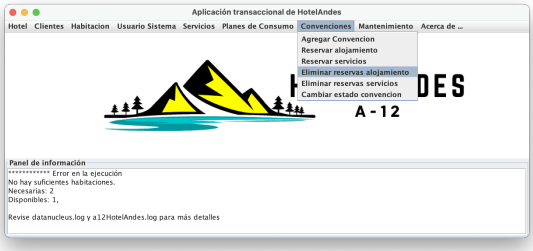


  Luego de que se agrega la convención, se hacen las reservas. Para ello el usuario debe ingresar el numero de habitaciones y los tipos respectivos o los servicios que desea reservar. Si el número de servicios o habitaciones disponibles no es suficiente, las reservas no se puede llevar a cabo:



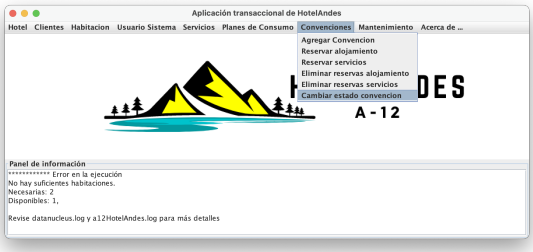
- **CANCELAR RESERVAS ASOCIADAS A UNA CONVENCIÓN**

  Para este requerimiento lo que se debe hacer es ingresar y eliminar las reservas que se quieran eliminar, de acuerdo con la convención. Esto mediante el siguiente patrón en el aplicativo:



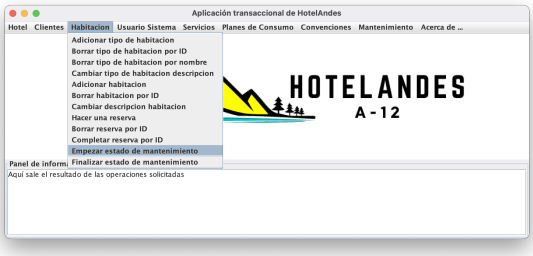
- **REGISTRAR EL FIN DE UNA CONVENCIÓN**

  Para este requerimiento el usuario debe ir a la convención que quiere registrar su fin y marcarlo. Esto se hace mediante el siguiente orden:



- **REGISTRAR LA ENTRADA A MANTENIMIENTO DE ALOJAMIENTOS O SERVICIOS DEL HOTEL**

  Para este requerimiento lo que se hace es que el usuario ingresa el id de la habitación (o del servicio) que estará en mantenimiento. Esto impide que se hagan reservas a esta habitación o a este servicio.



- **REGISTRAR EL FIN DEL MANTENIMIENTO DE ALOJAMIENTOS O SERVICIOS DEL HOTEL**

  Para este requerimiento se busca la habitación o el servicio que el usuario ingrese y se le termina el estado de mantenimiento. Esto permite que se hagan reservas nuevamente.

3.2.2 Requerimientos funcionales de consulta

- **ANALIZAR LA OPERACIÓN DE HOTELANDES**
  Nota: Con el fin de no saturar el documento para esta consulta en específico se muestran solo 2 casos desarrollados.

  Para este requerimiento se tomaron en cuenta las tablas de reservas de servicio y habitación, las tablas de servicio y habitación y de tipo de servicio y habitación para poder relacionar los tipos de servicio/habitación con sus reservas y obtener información sobre el tipo a su vez. El requerimiento pide que se consulten las fechas de mayor demanda con base en una unidad de tiempo (que pueden ser meses o semanas en este caso), un tipo de servicio e informar los de mayor y menor demanda. Teniendo esto en cuenta se hizo 4 consultar por tipo de habitación y tipo de servicio, es decir 8 en total. Esto, debido a que esta consulta se bifurca en las cantidad de unidades de tiempo sobre las que se quiere trabajar, en este caso 2. Por cada consulta por unidad de tiempo es necesario hacer 2 subconsultas, una para hallar los de mayor demanda y otra los de menor. Por ejemplo, esta consulta retorna la información de las 3 semanas del año en que más se solicitaron habitaciones del tipo especificado:

```
SELECT *
FROM
```

```
    (SELECT T.NOMBRE, to_number(to_char(to_date(HR.DIAENTRADA),'WW'))
        SEMANA, COUNT(H.TIPOHABITACION) NUM_USOS
    FROM HTA_RESERVA_HABITACION HR, HTA_HABITACION H, HTA_TIPO_HABITACION
         T
    WHERE H.TIPOHABITACION = 'TIPOHAB'
        AND T.ID = H.TIPOHABITACION
    GROUP BY T.NOMBRE, to_number(to_char(to_date(HR.DIAENTRADA),'WW'))
    ORDER BY NUM_USOS DESC)
FETCH NEXT 3 ROWS ONLY;
```

La parte más desafiante fue extraer de la fecha el número de la semana del año, puesto que lo demás fue aplicar el filtro correspondiente al número de habitación y escoger las 3 primeras en orden descendente. Un posible resultado de la anterior consulta es el siguiente:

NOMBRE	SEMANA	NUM_USOS
Suite Presidencial	1	6
Suite Presidencial	52	4
Suite Presidencial	17	2

Mientras que las semanas de menor concurrencia para la suite presidencial serían:

NOMBRE	SEMANA	NUM_USOS
Suite Presidencial	15	2
Suite Presidencial	12	2
Suite Presidencial	47	2

Para obtener esta misma información del tipo de servicio fue necesario (como se mencionó anteriormente) crear una nueva tabla tipoServicio y agregar un nuevo atributo a la tabla servicio. Para obtener el consumo efectivo de los servicios se utilizó la tabla reserva servicio por 2 razones. Primero, esta solo registra consumos efectuados, pues en caso de ser cancelada, la reserva es borrada del sistema. Segundo, permite acceder a una mayor variedad de servicios, sobre todo aquellos que no tienen costo. Puesto que el registro de su uso no se hace a través de consumos pagos, sino de reservas. Con respecto a las consultas de tipo de habitación surgió una dificultad adicional para extraer información sobre la fecha y es que la tabla reserva de servicios utiliza un timestamp y no un date. Esto causó un paso adicional y fue convertir un timestamo a un date. La consulta que obtiene las meses de menor demanda según el tipo de servicio es la siguiente:

```
SELECT *
FROM (
    SELECT T.NOMBRE, EXTRACT(MONTH FROM to_date(to_char(cast(RS.
        FECHAINICIO as date),'DD-MM-YYYY'))) MES, COUNT(S.TIPOSERVICIO)
        NUM_USOS
    FROM HTA_RESERVA_DE_SERVICIO RS, HTA_SERVICIO S, HTA_TIPO_SERVICIO T
    WHERE S.TIPOSERVICIO = T.ID
        AND RS.IDSERVICIO = S.ID
        AND T.ID = 'TIPOSERVICIO'
```

```
    GROUP BY T.NOMBRE, EXTRACT(MONTH FROM to_date(to_char(cast(RS.
        FECHAINICIO as date),'DD-MM-YYYY')))
    ORDER BY NUM_USOS ASC)
FETCH NEXT 3 ROWS ONLY;
```

Aquí extraer el número de mes de la fecha fue mucho más fácil que sacar el número de semana. Además de esto, se hacen los mismos filtros que en el anterior caso y se organizan de manera ascendente o descendente para obtener el resultado filtrado. El resultado para los meses de mayor demanda para el tipo de servicio 2 (correspondiente al gimnasio) se muestra a continuación.

NOMBRE	MES	NUM_USOS
Gimnasio	12	4
Gimnasio	4	4
Gimnasio	3	4

Mientras que los de menor demanda

NOMBRE	MES	NUM_USOS
Gimnasio	2	1
Gimnasio	1	1
Gimnasio	10	1

- **ENCONTRAR LOS BUENOS CLIENTES** Para encontrar los buenos clientes se realizaron 2 filtros principales. Primero se halló los clientes que se habían hospedado por más de 2 semanas en el hotel durante el último año de funcionamiento de hotelandes. Para esto, se tomaron en cuenta las tablas de cliente (para obtener su información) y reserva habitación que contiene la fecha de llegada y salida del cliente:

```
SELECT C1.TIPODOC, C1.NUMDOC, C1.NOMBRE
FROM
    (SELECT C.TIPODOC, C.NUMDOC, C.NOMBRE, SUM(TRUNC(RH.DIASALIDA)-TRUNC(
        RH.DIAENTRADA)) dias
    FROM HTA_CLIENTE C, HTA_RESERVA_HABITACION RH
    WHERE RH.NUMDOCCLIENTE = C.NUMDOC
        AND RH.TIPODOCCLIENTE = C.TIPODOC
        AND (TRUNC(SYSDATE)-(DIAENTRADA))<365
    GROUP BY C.TIPODOC,C.NUMDOC, C.NOMBRE) C1
WHERE dias > 14;
```

De esta manera, se hayan los clientes que se hospedaron más de una semana en el último año de hotelandes. El segundo filtro a realizar era hallaar los clientes que hubiesen consumido

más de 15 millones en sus estadías durante el último año. Para esto, se tomaron en cuenta las mismas tablas, puesto que la tabla reserva da cuenta del gasto total del cliente durante su estadía:

```
    SELECT C.TIPODOC, C.NUMDOC, C.NOMBRE
    FROM
        (SELECT C.TIPODOC, C.NUMDOC, C.NOMBRE, SUM(RH.CUENTA) GASTO
        FROM HTA_CLIENTE C, HTA_RESERVA_HABITACION RH
        WHERE RH.NUMDOCCLIENTE = C.NUMDOC
            AND RH.TIPODOCCLIENTE = C.TIPODOC
            AND (TRUNC(SYSDATE)-TRUNC(DIAENTRADA))<365
        GROUP BY C.TIPODOC, C.NUMDOC, C.NOMBRE) C
    WHERE C.GASTO > 15000000;
```

Finalmente, ya que ambas tablas cuentan con la misma información del cliente se hizo un union de ambos resultados y se obtuvo todos los buenos clientes del hotel.

```
    SELECT *
FROM (
    (SELECT C1.TIPODOC, C1.NUMDOC, C1.NOMBRE
    FROM
        (SELECT C.TIPODOC, C.NUMDOC, C.NOMBRE, SUM(TRUNC(RH.DIASALIDA)-TRUNC(
            RH.DIAENTRADA)) dias
        FROM HTA_CLIENTE C, HTA_RESERVA_HABITACION RH
        WHERE RH.NUMDOCCLIENTE = C.NUMDOC
            AND RH.TIPODOCCLIENTE = C.TIPODOC
            AND (TRUNC(SYSDATE)-(DIAENTRADA))<365
        GROUP BY C.TIPODOC,C.NUMDOC, C.NOMBRE) C1
    WHERE dias > 14)
    UNION
    (SELECT C.TIPODOC, C.NUMDOC, C.NOMBRE
    FROM
        (SELECT C.TIPODOC, C.NUMDOC, C.NOMBRE, SUM(RH.CUENTA) GASTO
        FROM HTA_CLIENTE C, HTA_RESERVA_HABITACION RH
        WHERE RH.NUMDOCCLIENTE = C.NUMDOC
            AND RH.TIPODOCCLIENTE = C.TIPODOC
            AND (TRUNC(SYSDATE)-TRUNC(DIAENTRADA))<365
        GROUP BY C.TIPODOC, C.NUMDOC, C.NOMBRE) C
    WHERE C.GASTO > 15000000));
```

Tome los siguientes datos como de ejemplo, tenga en cuenta que el formato de la fecha (AAAA,MM,DD):

Reserva Habitación							
idReserva	idHabitacion	numDocCliente	tipoDocCliente	diaEntrada	diaSalida	completada	cuenta
25	13	1000654218	CC	2016,04,12	2016,04,17	Y	0
26	16	1000985283	CC	2021,11,25	2021,12,03	Y	0
27	18	1000416291	CE	2020,08,05	2020,08,12	N	0
65	15	1000985283	CC	2020,12,28	2020,01,04	Y	0
66	19	1000768302	CC	2020,12,28	2020,01,04	Y	32000000
67	14	1000438921	CC	2022,03,23	2022,03,23	Y	7000000
68	14	1000438921	CC	2022,04,28	2022,05,05	Y	8005000
69	17	1000654218	CC	2022,01,06	2022,01,25	Y	3000000
70	13	1000416291	CE	2022,01,06	2022,01,07	Y	3000000
71	13	1000416291	CE	2022,01,06	2022,01,07	Y	3000000

Para estos, el resultado es el siguiente:

TIPODOC	NUMDOC	NOMBRE
CC	1000438921	Alejandra Villamil
CC	1000654218	Juan Diego Pardo

Como se puede observar a pesar de que Sergio Yepes, por haberse hospedado en el hotel más de 2 semanas y Esteban Gonzalez, por haber consumido más de 15 millones en el hotel, podrían ser considerados buenos clientes no aparecen en la tabla ya que sus estadías superan el último año de funcionamiento de hotelandes.

- **ENCONTRAR LOS SERVICIOS QUE NO TIENEN MUCHA DEMANDA** Para encontrar los servicios que no tienen mucha demanda se utilizaron las tablas de servicio y reserva de servicio. La clave para aplicar el filtro yace sobre la tabla reserva servicio, ya que primero se realiza un count del número de reservas agrupado por semana, teniendo en cuenta solo las reservas del último año, y posteriormente se saca un promedio de la cantidad de reservas por semana que tienen los servicios. Finalmente, se seleccionan los servicios cuyo promedio sea menor a 3 reservas semanales:

```
SELECT ID
FROM(
    SELECT A.ID, AVG(RESERVAS) PROM_SEMANAL
    FROM
        (SELECT S.ID, TRUNC(RS.FECHAINICIO,'IW') WEEK, COUNT(RS.
            IDSERVICIO) RESERVAS
        FROM HTA_SERVICIO S, HTA_RESERVA_DE_SERVICIO RS
        WHERE S.ID = RS.IDSERVICIO
        AND (TRUNC(SYSDATE)-TRUNC(RS.FECHAINICIO))<365
        GROUP BY S.ID, TRUNC(RS.FECHAINICIO,'IW'))A
    GROUP BY A.ID)
WHERE PROM_SEMANAL < 3;
```

Tomemos como ejemplo las reservas asociadas al servicio 31:

Reservas de Servicio				
id	idReservaHabitacion	idServicio	fechaInicio	fechaFin
103	26	31	2021,12,01 - 11:00:00	2021,12,0 - 12:00:00
104	26	31	2022,03,12 - 11:00:00	2022,03,12 - 12:00:00
105	26	31	2022,03,12 - 12:00:00	2022,03,12 - 13:00:00
106	26	31	2022,03,12 - 13:00:00	2022,03,12 - 14:00:00

El resultado con nuestros datos de prueba fue el siguiente:

ID	PROM_SEMANAL
31	2

Esto se debe a que se toman en cuenta la semana del 12 de Marzo de 2022, en la que hubo 3 reservas y la semana 1 de diciembre de 2021 en la que solo hubo una reserva. Al promediarlas, el resultado es obtenido es menor a 3 y la base de datos retorna el servicio 31 como un servicio de poca demanda. Sin embargo, si se ignora el dato de la semana del 1 de diciembre la consulta no retorna el servicio 31.

ID	PROM_SE...

3.2.3 Requerimientos no funcionales

Para esta iteración el requerimiento no funcional que se debía asegurar era la **transaccionalidad**. Para esto cabe recalcar que como verificamos todos los elementos con los que debería cumplir una transacción (ACID) es posible afirmar que nuestro proyecto cumple con este requerimiento funcional. De igual modo, al estar utilizando la base de datos de Oracle esto se da por centado con este SMBD.

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