

ENTREGA 2 – SISTTRANS - INFORME

Wilmer Manuel Arévalo González - 202214720 - w.arevalo

Sara Sofía Cárdenas Rodríguez - 202214907 - ss.cardenas

Daniel Felipe Díaz Moreno – 202210733 - d.diazm

1. ANÁLISIS

1.1. MODELO UML

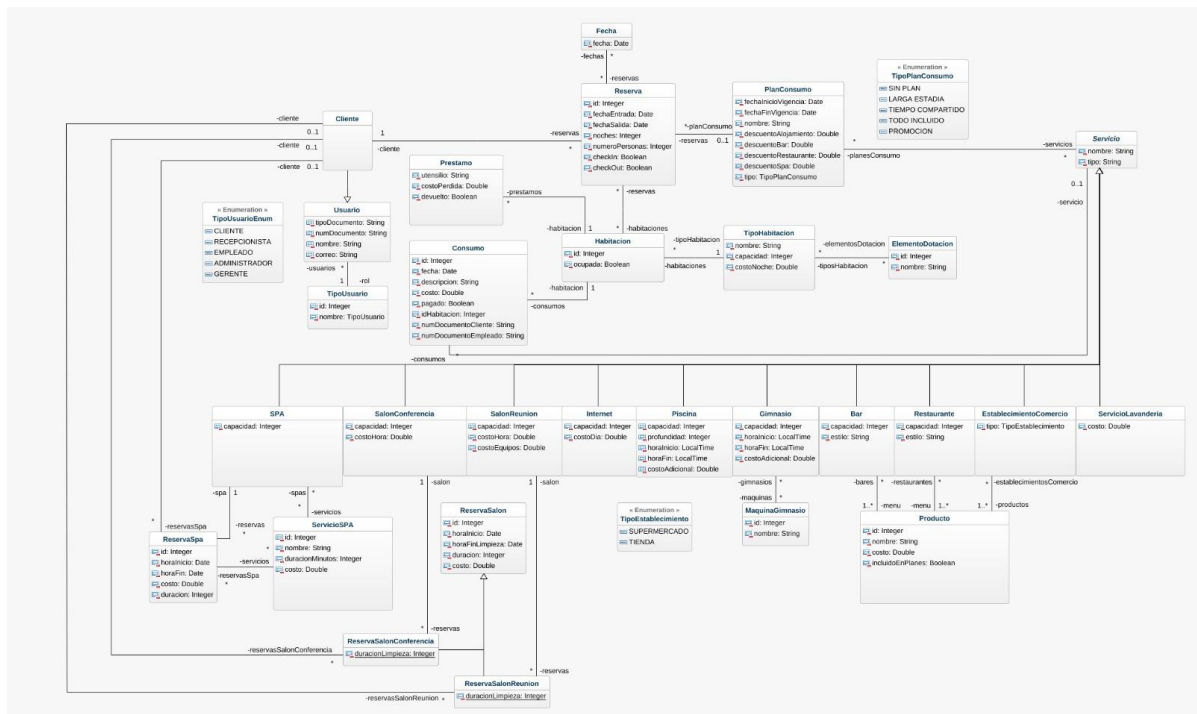
Se crearon asociaciones entre el usuario y más clases, ya que no todas las reservas apuntaban al cliente que las tenía (reservasSpa, reservasSalonConferencia, reservasSalonReunion)

Se creó una nueva clase llamada Fechas, con el fin de asociarse con las reservas y obtener el índice de ocupación de las habitaciones del hotel

A Consumos, se le añadió un atributo num_documento_empleado para apuntar al empleado que registró el consumo

A Servicios, se le añadió un atributo tipo para modelar los tipos de servicios mencionados en el enunciado

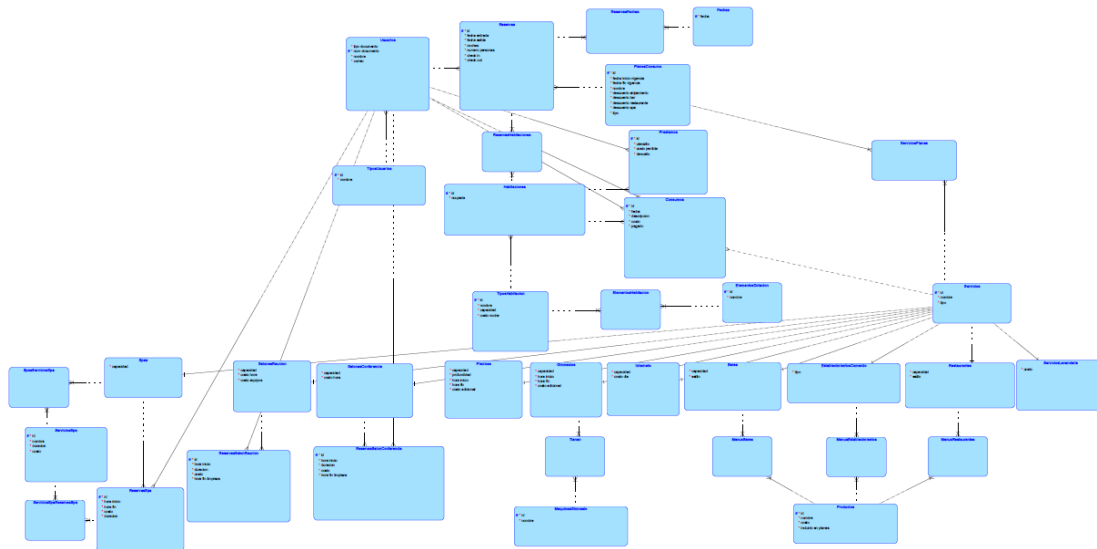
A ReservasSpa, se le añadió un atributo duración para facilitar la obtención de la duración de los servicios



Para obtener una resolución óptima, consulte el archivo UML.pdf ubicado en la carpeta "modelos" dentro del directorio "docs".

1.2. MODELO E/R

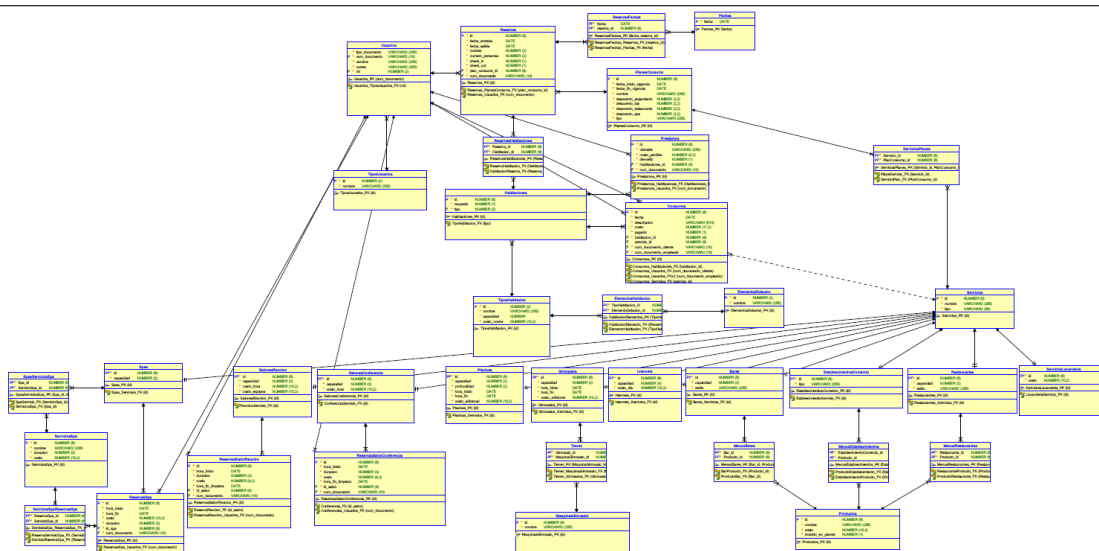
De igual manera, los cambios se reflejan en este modelo



Para obtener una resolución óptima, consulte el archivo modeloER.pdf ubicado en la carpeta "modelos" dentro del directorio "docs".

1.3. MODELO RELACIONAL

De igual manera, los cambios se reflejan en este modelo



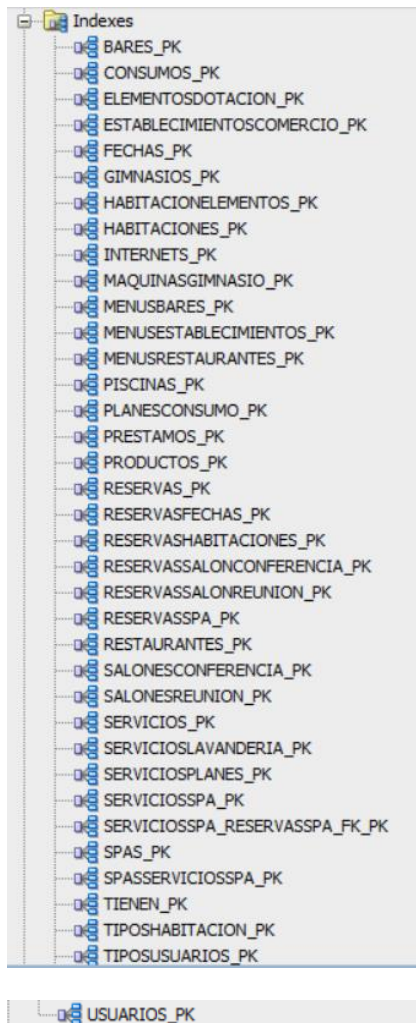
Para obtener una resolución óptima, consulte el archivo modeloRelacional.pdf ubicado en la carpeta "modelos" dentro del directorio "docs"

2. DISEÑO DE LA APP

2.1. ÍNDICES

2.1.1. ÍNDICES DE ORACLE

Oracle creó 37 índices para modelar las 37 llaves primarias de todas las tablas.



Cabe resaltar que, para las tablas de asociación, como hay dos columnas en sus llaves primarias, se generaron índices compuestos sobre ambas columnas. Por ejemplo, en el menú de los bares

Actions...						
INDEX_OWNER	INDEX_NAME	TABLE_OWNER	TABLE_NAME	COLUMN_NAME	COLUMN_POSITION	DESCEND
1 ISIS2304E23202320	MENUSBARES	PK ISIS2304E23202320	MENUSBARES	BAR ID	1	ASC
2 ISIS2304E23202320	MENUSBARES	PK ISIS2304E23202320	MENUSBARES	PRODUCTO ID	2	ASC

Entonces, estos índices fueron creados ya que corresponden a las PK de todas las tablas, y evidentemente aumentan la velocidad de todos los requerimientos, dado que las PK son columnas susceptibles al uso de joins y a consultas de todo tipo. Además, cuentan con una altísima selectividad

2.1.2. ÍNDICES CREADOS

Se crearon 25 índices, especialmente sobre atributos usados para join, como FKs, y sobre columnas frecuentemente consultadas para rangos, como los costos o fechas

```
CREATE INDEX reservas_fecha_entrada ON reservas(fecha_entrada);
CREATE INDEX reservas_fecha_salida ON reservas(fecha_salida);
CREATE INDEX reservas_noches ON reservas(noches);
CREATE INDEX reservas_num_documento ON reservas(num_documento);
CREATE INDEX consumos_fecha ON consumos(fecha);
CREATE INDEX consumos_costo ON consumos(costo);
CREATE INDEX consumos_servicio_id ON consumos(servicio_id);
CREATE INDEX consumos_habitacion_id ON consumos(habitacion_id);
CREATE INDEX consumos_num_documento_cliente ON consumos(num_documento_cliente);
CREATE INDEX consumos_num_documento_empleado ON consumos(num_documento_empleado);
CREATE INDEX servicios_tipo ON servicios(tipo);
CREATE INDEX salonesReunion_costo_hora ON salonesReunion(costo_hora);
CREATE INDEX salonesConferencia_costo_hora ON salonesConferencia(costo_hora);
CREATE INDEX piscinas_costo_adicional ON piscinas(costo_adicional);
CREATE INDEX gimnasios_costo_adicional ON gimnasios(costo_adicional);
CREATE INDEX internets_costo_dia ON internets(costo_dia);
CREATE INDEX productos_costo ON productos(costo);
CREATE INDEX serviciosLavanderia_costo ON serviciosLavanderia(costo);
```

```
CREATE INDEX serviciosSpa_costo ON serviciosSpa(costo);
CREATE INDEX reservasSalonReunion_duracion ON reservasSalonReunion(duracion);
CREATE INDEX reservasSalonReunion_num_documento ON reservasSalonReunion(num_documento);
CREATE INDEX reservasSalonConferencia_duracion ON reservasSalonConferencia(duracion);
CREATE INDEX reservasSalonConferencia_num_documento ON reservasSalonConferencia(num_documento);
```

```
CREATE INDEX reservasSpa_duracion ON reservasSpa(duracion);
CREATE INDEX reservasSpa_num_documento ON reservasSpa(num_documento);
```

RFC1: DINERO RECOLECTADO POR SERVICIOS EN CADA HABITACIÓN EN EL ÚLTIMO AÑO.

Se crearon tres índices en Consumos, uno para la fecha, otro para el costo y otro para el id de la habitación

Se desearía crear un índice de hash en id_habitación, dados los valores únicos, pero no fue posible. Se optó por B+ secundario

Para los demás, se creó un índice B+ de tipo secundario, para acceder rápidamente a rangos y a valores de estos atributos

Bajo selectividad para la fecha, hay 730 días entre 2022 y 2023, por lo que la alta selectividad es del 0.13 %

Bajo selectividad para el costo, hay 1 E10 posibles valores en pesos colombianos, pero normalmente oscilan entre 0 y 1 millón de pesos, por lo que la alta selectividad es del 0.0001 %

Bajo selectividad para id_habitación, como hay más de 1k habitaciones, la selectividad es menor que 0.001%

RFC2: 20 SERVICIOS MÁS POPULARES

Nuevamente se usa el índice de fecha en consumo, para evaluarlo entre dos fechas

RFC3: ÍNDICE DE OCUPACIÓN DE CADA UNA DE LAS HABITACIONES DEL HOTEL

Se crearon tres índices en Reservas, uno para la fecha de entrada, otro para la fecha de salida y otro para la cantidad de noches

Se creó un índice B+ de tipo secundario, para acceder rápidamente a rangos y a valores de estos atributos

La selectividad para las fechas es la misma

Para la cantidad de noches, se tiene mínimo una noche y máximo 21, por lo que la selectividad es del 4.76%

RFC4: SERVICIOS QUE CUMPLEN CON CIERTA CARACTERÍSTICA

Precio

Se crearon ocho índices en las tablas que heredan de servicios (Piscinas, Gimnasios, Internets, Salones, ServiciosLavanderia), para atributos que modelan precios (costo, costo_hora, costo_dia, costo_adicional)

Para los servicios sin costo asociado, se crearon dos índices sobre productos y serviciosSpa sobre la columna costo

Se creó un índice B+ de tipo secundario, para acceder rápidamente a rangos y a valores de estos atributos

Nuevamente la selectividad para costos es la misma

Empleado

Se creó un índice sobre la columna num_documento_empleado en Consumos

Se desearía crear un índice de hash, dados los valores únicos, pero no fue posible. Se optó por B+ secundario

Bajo selectividad, como hay más de 1k empleados, la selectividad es menor que 0.001%

Tipo

Se creó un índice sobre la columna tipo en Servicios

Se desearía crear un índice de hash, dados los valores únicos, pero no fue posible. Se optó por B+ secundario

Bajo selectividad, como hay 10 tipos, la selectividad es del 10%

RFC5: CONSUMO POR UN USUARIO DADO, EN UN RANGO DE FECHAS

Nuevamente se usa el índice de fecha en consumo, para evaluarlo entre dos fechas

RFC6: OPERACIÓN DE HOTELANDES

Se usaron los índices por defecto de Oracle, ya que eran solo PKs

RFC7: BUENOS CLIENTES

Se creó un índice en Reservas, para el num_documento

Se desearía crear un índice de hash, dados los valores únicos, pero no fue posible. Se optó por B+ secundario

Se crearon dos índices en Consumos, para servicio_id y para num_documento_cliente

Se desearía crear dos índices de hash, dados los valores únicos, pero no fue posible. Se optó por B+ secundario

Bajo selectividad de num_documento y num_documento_cliente, como hay más de 1k clientes, la selectividad es menor que 0.001%

Bajo selectividad de servicio_id, como hay más de 1k servicios, la selectividad es menor que 0.001%

RFC8: SERVICIOS QUE NO TIENEN MUCHA DEMANDA

Nuevamente se usa el índice de servicios_id en Consumos

RFC9: CONSUMO EN HOTELANDES

Nuevamente se usan los índices de consumos

RFC10: CONSUMO EN HOTELANDES v2

Nuevamente se usan los índices de consumos

RFC11: FUNCIONAMIENTO

Nuevamente se usan los índices en reservas para las fechas y los de consumos

RFC12: CLIENTES EXCELENTES

Se crearon cuatro índices en ReservasSalonReunión y ReservasSalonConferencia, para sus atributos num_documento y duración

En ReservasSpa, se crearon dos índices para num_documento y duración de igual forma

Se deseaba tres índices de hash para num_documento, dados los valores únicos, pero no fue posible. Se optó por B+ secundario

Se crearon tres índices B+ de tipo secundario, para acceder rápidamente a rangos y a valores de duración

Bajo selectividad de num_documento y num_documento_cliente, como hay más de 1k clientes, la selectividad es menor que 0.001%

Las duraciones en minutos rondan entre 15 y 720, por lo que la selectividad es del 0.13%

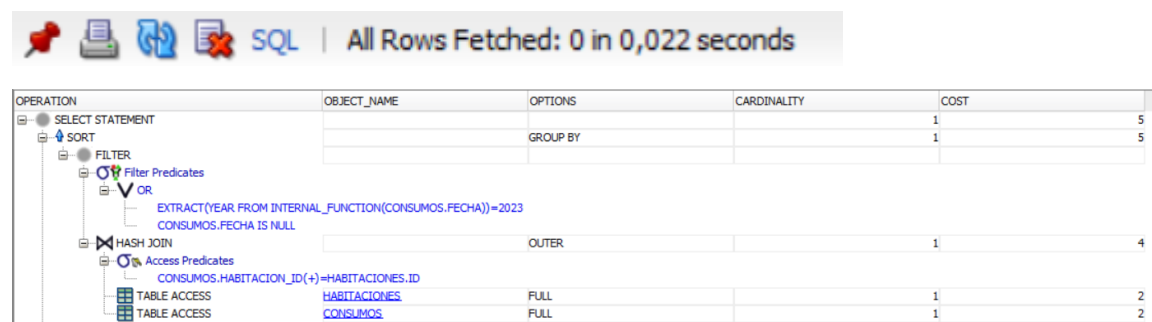
2.2. DISEÑO DE LAS CONSULTAS

Las sentencias SQL de consulta se encuentran en scripts/Sentencias

La distribución de los datos se hizo con grandes intervalos. Para fechas, se escogió todo el 2023. Para tipos, se escogió Bares y ServiciosLavandería. Se usaron usuarios predefinidos en la población de la anterior iteración. Se usaron costos no superiores a 100.000 pesos.

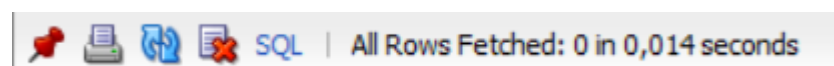
A continuación, se presentan los planes de consulta y tiempos de ejecución para la población reducida. Aun así, los planes deberían mantenerse en su estructura.

RFC1: DINERO RECOLECTADO POR SERVICIOS EN CADA HABITACIÓN EN EL ÚLTIMO AÑO.



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				5
SORT				5
FILTER		GROUP BY		5
Filter Predicates				
OR				
EXTRACT(YEAR FROM INTERNAL_FUNCTION(CONSUMOS.FECHA))=2023				
CONSUMOS.FECHA IS NULL				
HASH JOIN		OUTER	1	4
Access Predicates				
CONSUMOS.HABITACION_ID(+) = HABITACIONES.ID				
TABLE ACCESS	HABITACIONES	FULL	1	2
TABLE ACCESS	CONSUMOS	FULL	1	2

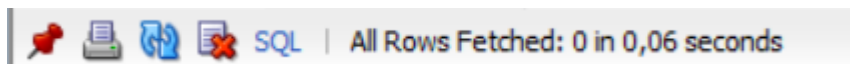
RFC2: 20 SERVICIOS MÁS POPULARES



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				5
SORT				5
FILTER		GROUP BY		5
Filter Predicates				
OR				
EXTRACT(YEAR FROM INTERNAL_FUNCTION(CONSUMOS.FECHA))=2023				
CONSUMOS.FECHA IS NULL				
HASH JOIN		OUTER	1	4
Access Predicates				
CONSUMOS.HABITACION_ID(+) = HABITACIONES.ID				
TABLE ACCESS	HABITACIONES	FULL	1	2
TABLE ACCESS	CONSUMOS	FULL	1	2

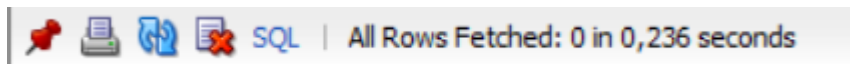
OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				5
SORT		ORDER BY		5
VIEW	SYS.NULL			4
Filter Predicates				20
from\$_subquery\$_004.rowlimit_\$\$_rownumber <= 20				20
WINDOW		SORT PUSHED RANK		1
Filter Predicates				4
ROW_NUMBER() OVER (ORDER BY COUNT(*) DESC) <= 20				1
HASH		GROUP BY		1
NESTED LOOPS				2
NESTED LOOPS				1
TABLE ACCESS	CONSUMOS	FULL		1
Filter Predicates				2
AND				1
CONSUMOS.FECHA >= TO_DATE('2023-01-01 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				1
CONSUMOS.FECHA <= TO_DATE('2023-12-31 23:59:00', 'yyyy-mm-dd hh24:mi:ss')				1
INDEX	SERVICIOS_PK	UNIQUE SCAN		0
Access Predicates				1
CONSUMOS.SERVICIO_ID = SERVICIOS.ID				1
TABLE ACCESS	SERVICIOS	BY INDEX ROWID		0

RFC3: ÍNDICE DE OCUPACIÓN DE CADA UNA DE LAS HABITACIONES DEL HOTEL

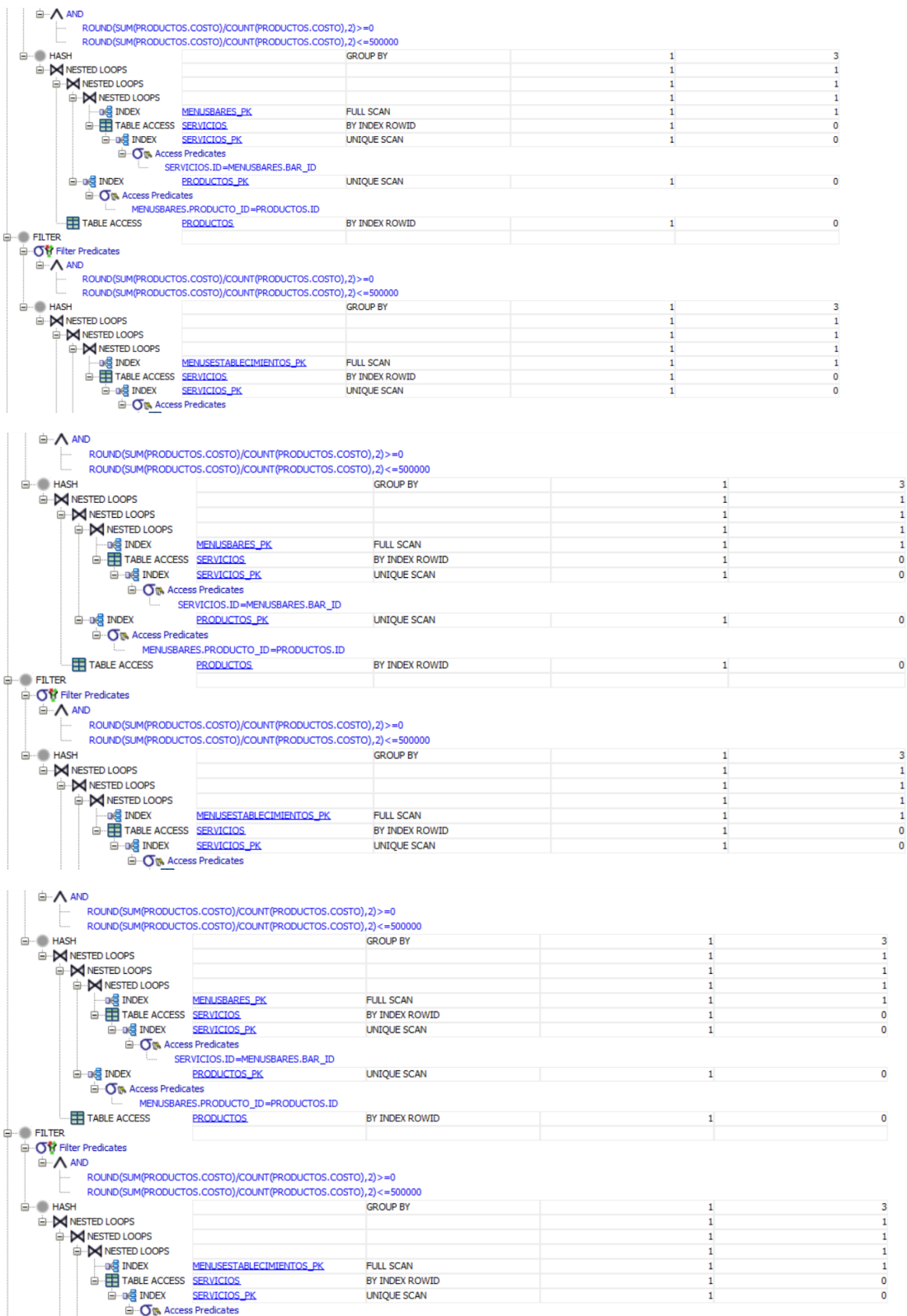


OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				4
SORT		GROUP BY		4
FILTER				1
Filter Predicates				1
OR				1
RESERVASHABITACIONES.HABITACION_ID IS NULL				1
RESERVAS.FECHA_SALIDA < (SELECT TO_DATE(TO_CHAR(SYSDATE @ \$ \$, 'YYYY') '-' '01-01', 'YYYY-MM-DD') + INTERVAL ' +01-00' YEAR (2) TO MONTH FROM SYS.DUAL DUAL)				1
NESTED LOOPS		OUTER		3
NESTED LOOPS		OUTER		3
TABLE ACCESS	HABITACIONES	FULL		2
INDEX	RESERVASHABITACIONES_PK	FULL SCAN		1
Access Predicates				1
HABITACIONES.ID = RESERVASHABITACIONES.HABITACION_ID (+)				1
Filter Predicates				1
HABITACIONES.ID = RESERVASHABITACIONES.HABITACION_ID (+)				1
TABLE ACCESS	RESERVAS	BY INDEX ROWID		0
INDEX	RESERVAS_PK	UNIQUE SCAN		0
Access Predicates				1
RESERVASHABITACIONES.RESERVA_ID = RESERVAS.ID (+)				1
FAST DUAL				2

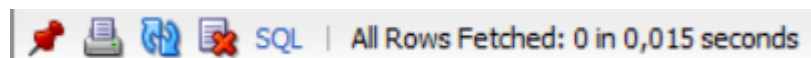
RFC4: SERVICIOS QUE CUMPLEN CON CIERTA CARACTERÍSTICA



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				6
INTERSECTION				1
SORT		UNIQUE		3
NESTED LOOPS				2
NESTED LOOPS				1
TABLE ACCESS	CONSUMOS	FULL		2
Filter Predicates				1
AND				1
CONSUMOS.NUM_DOCUMENTO_EMPLEADO = 428				1
CONSUMOS.FECHA >= TO_DATE('2023-01-01 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				1
CONSUMOS.FECHA <= TO_DATE('2023-12-31 00:00:00', 'yyyy-mm-dd hh24:mi:ss')				1
INDEX	SERVICIOS_PK	UNIQUE SCAN		0
Access Predicates				1
SERVICIOS.ID = CONSUMOS.SERVICIO_ID				1
TABLE ACCESS	SERVICIOS	BY INDEX ROWID		0
Filter Predicates				1
SERVICIOS.TIPO = 'ServicioLavanderia'				1
SORT		UNIQUE		3
FILTER				1
Filter Predicates				1
OR				1
AND				1
PISCINAS.COSTO_ADICIONAL >= 0				1
PISCINAS.COSTO_ADICIONAL <= 500000				1
AND				1
GIMNASIOS.COSTO_ADICIONAL >= 0				1
GIMNASIOS.COSTO_ADICIONAL <= 500000				1
AND				1
INTERNETS.COSTO_DTA >= 0				1

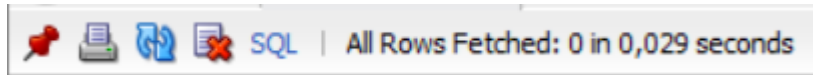


RFC5: CONSUMO POR UN USUARIO DADO, EN UN RANGO DE FECHAS

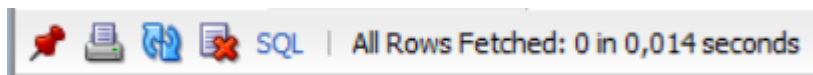


OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				
TABLE ACCESS	CONSUMOS	FULL	1	2
Filter Predicates				
AND				
NUM_DOCUMENTO_CLIENTE='105'				
FECHA>=TO_DATE('2023-01-01 00:00:00','yyyy-mm-dd hh24:mi:ss')				
FECHA<=TO_DATE('2023-12-31 23:59:00','yyyy-mm-dd hh24:mi:ss')				

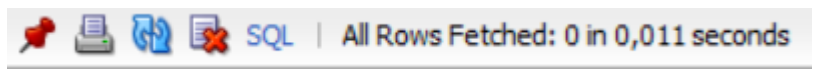
RFC6: OPERACIÓN DE HOTELANDES



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				
VIEW	SYS.NULL	ORDER BY		5
Filter Predicates				
from\$_subquery\$_008.rowlimit_\$\$_rownumber <= 20				
WINDOW		SORT PUSHED RANK		1
Filter Predicates				
ROW_NUMBER() OVER (ORDER BY COUNT(CASE WHEN RESERVASFECHAS.RESERVA_ID IS NOT NULL THEN 1 ELSE NULL END) DESC ,FECHAS.FECHA) <= 20				
GROUP BY NOSORT				4
NESTED LOOPS		OUTER		1
NESTED LOOPS		OUTER		3
NESTED LOOPS		OUTER		2
NESTED LOOPS		OUTER		2
INDEX	FECHAS_PK	FULL SCAN	1	1
INDEX	RESERVASFECHAS_PK	RANGE SCAN	1	1
Access Predicates				
FECHAS.FECHA=RESERVASFECHAS.FECHA(+)				
INDEX	RESERVAS_PK	UNIQUE SCAN	1	0
Access Predicates				
RESERVASFECHAS.RESERVA_ID=RESERVAS.ID(+)				
INDEX	RESERVASHABITACIONES_PK	RANGE SCAN	1	1
Access Predicates				
RESERVAS.ID=RESERVASHABITACIONES.RESERVA_ID(+)				

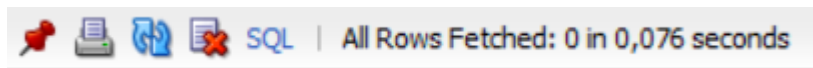


OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				
VIEW	SYS.NULL	ORDER BY		5
Filter Predicates				
from\$_subquery\$_008.rowlimit_\$\$_rownumber <= 20				
WINDOW		SORT PUSHED RANK		1
Filter Predicates				
ROW_NUMBER() OVER (ORDER BY COUNT(CASE WHEN RESERVASFECHAS.RESERVA_ID IS NOT NULL THEN 1 ELSE NULL END) ,FECHAS.FECHA) <= 20				
GROUP BY NOSORT				4
NESTED LOOPS		OUTER		1
NESTED LOOPS		OUTER		3
NESTED LOOPS		OUTER		2
NESTED LOOPS		OUTER		2
INDEX	FECHAS_PK	FULL SCAN	1	1
INDEX	RESERVASFECHAS_PK	RANGE SCAN	1	1
Access Predicates				
FECHAS.FECHA=RESERVASFECHAS.FECHA(+)				
INDEX	RESERVAS_PK	UNIQUE SCAN	1	0
Access Predicates				
RESERVASFECHAS.RESERVA_ID=RESERVAS.ID(+)				
INDEX	RESERVASHABITACIONES_PK	RANGE SCAN	1	1
Access Predicates				
RESERVAS.ID=RESERVASHABITACIONES.RESERVA_ID(+)				



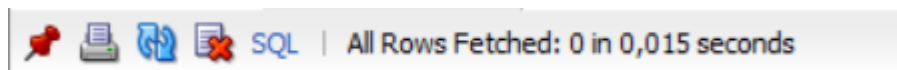
OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				
VIEW	SYS.NULL	ORDER BY		5
Filter Predicates				
from\$_subquery\$_002.rowlimit_\$\$_rownumber <= 20				
WINDOW		SORT PUSHED RANK		1
Filter Predicates				
ROW_NUMBER() OVER (ORDER BY COUNT(*) DESC) <= 20				
HASH		GROUP BY		4
TABLE ACCESS	CONSUMOS	FULL	1	2

RFC7: BUENOS CLIENTES



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				6
FILTER			1	
Filter Predicates				
OR				
SUM(CONSUMOS.COSTO)>15000000				
SUM(RESERVAS.NOCHES)>=14				
SORT				6
HASH JOIN		GROUP BY	1	
Access Predicates		OUTER	1	5
USUARIOS.NUM_DOCUMENTO=RESERVAS.NUM_DOCUMENTO(+)				
HASH JOIN		OUTER	1	3
Access Predicates				
USUARIOS.NUM_DOCUMENTO=CONSUMOS.NUM_DOCUMENTO_CLIENTE(+)				
INDEX	USUARIOS_PK	FULL SCAN	1	1
TABLE ACCESS	CONSUMOS	FULL	1	2
TABLE ACCESS	RESERVAS	FULL	1	2

RFC8: SERVICIOS QUE NO TIENEN MUCHA DEMANDA



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				6
SORT		ORDER BY		6
FILTER			1	
Filter Predicates				
ROUND(COUNT(CONSUMOS.SERVICIO_ID)/TO_NUMBER(TO_CHAR(SYSDATE@!, 'VVV'), 2) < 3				
HASH		GROUP BY	1	6
HASH JOIN		OUTER	1	4
Access Predicates				
CONSUMOS.SERVICIO_ID(+) = SERVICIOS.ID				
TABLE ACCESS	SERVICIOS	FULL	1	2
TABLE ACCESS	CONSUMOS	FULL	1	2

RFC9: CONSUMO EN HOTELANDES

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				2
SORT		ORDER BY		2
FILTER				
Filter Predicates				
TO_DATE(FECHAFINAL, 'YYYY-MM-DD') >= TO_DATE(FECHAINICIAL, 'YYYY-MM-DD')				
NESTED LOOPS			1	2
NESTED LOOPS			1	2
NESTED LOOPS			1	2
TABLE ACCESS	CONSUMOS	FULL	1	2
Filter Predicates				
AND				
C.FECHA >= TO_DATE(FECHAINICIAL, 'YYYY-MM-DD')				
C.FECHA <= TO_DATE(FECHAFINAL, 'YYYY-MM-DD')				
TABLE ACCESS	USUARIOS	BY INDEX ROWID	1	0
INDEX	USUARIOS_PK	UNIQUE SCAN	1	0
Access Predicates				
US.NUM_DOCUMENTO=C.NUM_DOCUMENTO_CLIENTE				
INDEX	SERVICIOS_PK	UNIQUE SCAN	1	0
Access Predicates				
S.ID=C.SERVICIO_ID				
TABLE ACCESS	SERVICIOS	BY INDEX ROWID	1	0
Filter Predicates				
S.TIPO=SERVICIO				

RFC10: CONSUMO EN HOTELANDES v2

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	4
SORT		ORDER BY	1	4
FILTER				
Filter Predicates				
S.ID IS NULL				
NESTED LOOPS		OUTER	1	4
FILTER				
Filter Predicates				
OR				
C.FECHA IS NULL				
AND				
C.FECHA >= TO_DATE(FECHA INICIAL, 'YYYY-MM-DD')				
C.FECHA <= TO_DATE(FECHA FINAL, 'YYYY-MM-DD')				
HASH JOIN		OUTER	1	4
Access Predicates				
US.NUM_DOCUMENTO=C.NUM_DOCUMENTO_CLIENTE(+)				
TABLE ACCESS	USUARIOS	FULL	1	2
TABLE ACCESS	CONSUMOS	FULL	1	2
TABLE ACCESS	SERVICIOS	BY INDEX ROWID	1	0
Filter Predicates				
S.TIPO(+) =:SERVICIO				
INDEX	SERVICIOS_PK	UNIQUE SCAN	1	0
Access Predicates				
S.ID(+) =C.SERVICIO_ID				

RFC11: FUNCIONAMIENTO

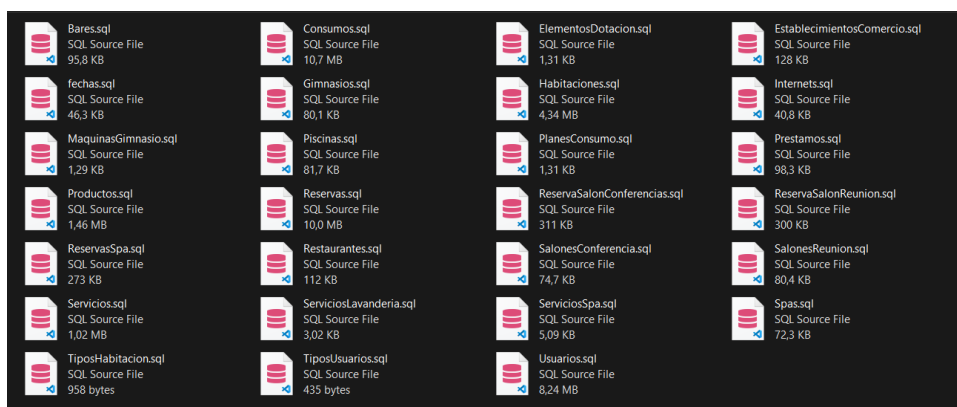
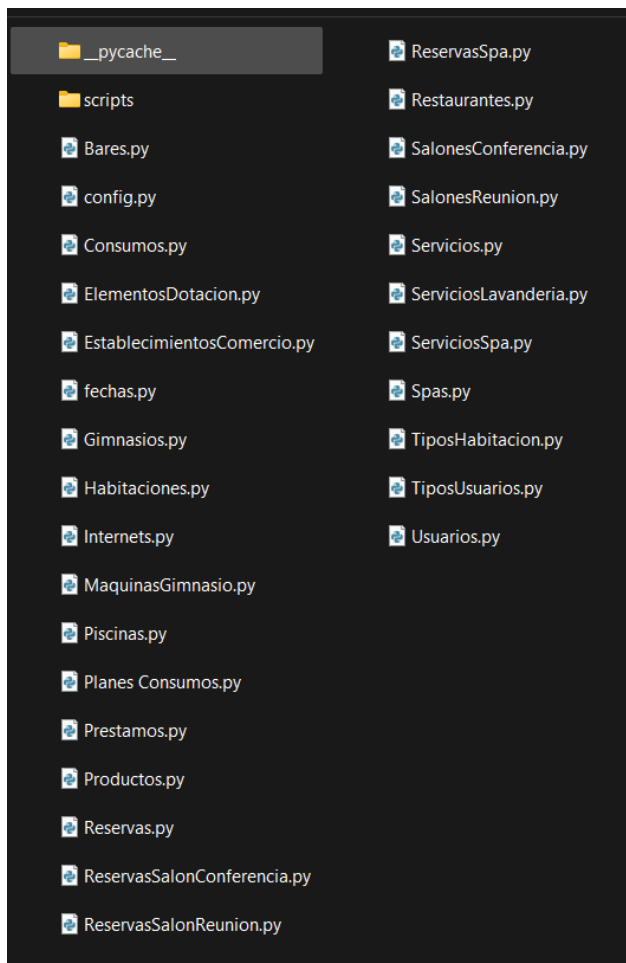
OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	3
FILTER				
Filter Predicates				
COUNT(TABLA.NUM_DOCUMENTO)=4				
HASH		GROUP BY	1	3
VIEW			1	3
HASH		GROUP BY	1	3
NESTED LOOPS			1	2
NESTED LOOPS			1	2
TABLE ACCESS	RESERVAS	FULL	1	2
Filter Predicates				
EXTRACT(YEAR FROM INTERNAL_FUNCTION(RESERVAS.FECHA_SALIDA))=2023				
INDEX	USUARIOS_PK	UNIQUE SCAN	1	0
Access Predicates				
USUARIOS.NUM_DOCUMENTO=RESERVAS.NUM_DOCUMENTO				
TABLE ACCESS	USUARIOS	BY INDEX ROWID	1	0

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	3
SORT		ORDER BY	1	3
HASH		GROUP BY	1	3
NESTED LOOPS			1	1
NESTED LOOPS			1	1
NESTED LOOPS			1	1
INDEX	RESERVASHABITACIONES_PK	FULL SCAN	1	1
TABLE ACCESS	HABITACIONES	BY INDEX ROWID	1	0
INDEX	HABITACIONES_PK	UNIQUE SCAN	1	0
Access Predicates				
RESERVASHABITACIONES.HABITACION_ID=HABITACIONES.ID				
INDEX	RESERVAS_PK	UNIQUE SCAN	1	0
Access Predicates				
RESERVAS.ID=RESERVASHABITACIONES.RESERVA_ID				
TABLE ACCESS	RESERVAS	BY INDEX ROWID	1	0
Filter Predicates				
EXTRACT(YEAR FROM INTERNAL_FUNCTION(RESERVAS.FECHA_ENTRADA))=2023				

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	4
SORT		ORDER BY	1	4
HASH		GROUP BY	1	4
NESTED LOOPS			1	2
NESTED LOOPS			1	2
TABLE ACCESS	CONSUMOS	FULL	1	2
Filter Predicates				
EXTRACT(YEAR FROM INTERNAL_FUNCTION(CONSUMOS.FECHA))=2023				
INDEX	SERVICIOS_PK	UNIQUE SCAN	1	0
Access Predicates				
CONSUMOS.SERVICIO_ID=SERVICIOS.ID				
TABLE ACCESS	SERVICIOS	BY INDEX ROWID	1	0

RFC12: CLIENTES EXCELENTES

No obstante, las relaciones de un tamaño menor a 50 tuplas fueran escritas a mano, usando valores razonables



Clases de asociación

En el caso de las asociaciones donde uno de los extremos tiene una cardinalidad de "muchos", también se elige aleatoriamente la cantidad de elementos relacionados, asegurando que esta cantidad esté dentro de un rango válido, respetando las restricciones de los tipos de datos definidas en los modelos relacionales

Un caso especial de clases de asociación fueron los menús de Bares, Establecimientos de Comercio y Restaurantes, donde se distribuyeron los productos.

Para esto, el menú de un bar serían tuplas contiguas en memoria, y los menús de dos bares contiguos tendrán productos en común. Entonces, un bar podría tener un menú de 50 productos, y 20 de esos productos también estarán en el menú de otro bar, para asemejarse a la vida real.

