

Iteración 2

Análisis:

- Los diagramas no cambian.

Diseño de la aplicación:

- Índices y consultas:

- RFC1

```
SELECT h.id AS habitacion_id,  
       COALESCE(SUM(c.costo), 0) AS dinero_recolectado  
FROM  
       habitaciones h  
LEFT JOIN  
       consumos c ON h.id = c.habitaciones_id  
WHERE  
       c.fecha_consumo >= ADD_MONTHS(CURRENT_DATE, -12)  
GROUP BY  
       h.id;
```

No es necesario añadir un índice extra ya que lo único crítico que usa es el id de la habitación.

- RFC2

```
SELECT  
       s.tipo AS servicio,  
       COUNT(c.tiposervicio_tipo) AS cantidad_consumos  
FROM  
       consumos c  
INNER JOIN  
       tiposervicio s ON c.tiposervicio_tipo = s.tipo  
WHERE  
       c.fecha_consumo >= TRUNC(SYSDATE - INTERVAL '15' MONTH) --  
Período de 6 meses  
GROUP BY  
       s.tipo  
ORDER BY  
       cantidad_consumos DESC  
FETCH FIRST 20 ROWS ONLY;
```

- RFC3

```

WITH ReservasEnUltimoAnio AS (
    SELECT
        r.habitaciones_id,
        GREATEST(rs.fecha_inicio, (SYSDATE - INTERVAL '1' YEAR)) AS
fecha_entrada,
        LEAST(rs.fecha_salida, SYSDATE) AS fecha_salida
    FROM
        reservan r
    INNER JOIN
        reservas rs ON r.reservas_id = rs.id
    WHERE
        rs.fecha_inicio <= SYSDATE
        AND rs.fecha_salida >= (SYSDATE - INTERVAL '1' YEAR)
)
SELECT
    h.id AS habitacion_id,
    ROUND(COALESCE(SUM(TO_NUMBER(fecha_salida - fecha_entrada) +
1), 0)) AS dias_ocupados_en_ultimo_anio,
    ROUND(COALESCE((SUM(TO_NUMBER(fecha_salida - fecha_entrada) +
1) / 365) * 100, 0)) AS porcentaje_ocupacion_en_ultimo_anio
FROM
    habitaciones h
LEFT JOIN
    ReservasEnUltimoAnio ro ON h.id = ro.habitaciones_id
GROUP BY
    h.id;

```

- RFC5

```

SELECT
    u.num_documento AS numero_documento,
    u.nombre,
    SUM(c.costo) AS suma_consumos
FROM
    reservas r
INNER JOIN
    habitaciones h ON r.id = h.id
INNER JOIN
    consumos c ON h.id = c.habitaciones_id
INNER JOIN
    usuarios u ON r.usuarios_num_documento = u.num_documento AND
r.usuarios_tipo_documento = u.tipo_documento
WHERE

```

u.num_documento = '1765432' -- Reemplaza con el número de documento
del usuario que deseas buscar

AND c.fecha_consumo BETWEEN '01/01/2023' AND '30/12/2023' --
Reemplaza con el rango de fechas deseado

GROUP BY

u.num_documento, u.nombre;

- RFC6

WITH FechasReservas AS (

SELECT DISTINCT fecha AS fecha_reserva

FROM (

SELECT fecha_inicio AS fecha FROM reservas

UNION ALL

SELECT fecha_salida AS fecha FROM reservas))

SELECT fr.fecha_reserva, COUNT(r.id) AS reservas_activas

FROM FechasReservas fr

JOIN reservas r ON fr.fecha_reserva BETWEEN r.fecha_inicio AND
r.fecha_salida

GROUP BY fr.fecha_reserva

ORDER BY reservas_activas DESC

FETCH FIRST 10 ROWS ONLY;

--- B)

WITH FechasConsumos AS (

SELECT DISTINCT fecha_consumo AS fecha

FROM consumos)

SELECT fc.fecha, SUM(c.costos) AS costos_consumos, COUNT(c.costos) AS
consumos_realizados

FROM FechasConsumos fc

JOIN consumos c ON fc.fecha = c.fecha_consumo

GROUP BY fc.fecha

ORDER BY costos_consumos DESC

FETCH FIRST 10 ROWS ONLY;

--- C

WITH FechasReservas AS (

SELECT DISTINCT fecha AS fecha_reserva

FROM (

SELECT fecha_inicio AS fecha FROM reservas

UNION ALL

SELECT fecha_salida AS fecha FROM reservas))

SELECT fr.fecha_reserva, COUNT(r.id) AS reservas_activas

```

FROM FechasReservas fr
JOIN reservas r ON fr.fecha_reserva BETWEEN r.fecha_inicio AND
r.fecha_salida
GROUP BY fr.fecha_reserva
ORDER BY reservas_activas ASC
FETCH FIRST 10 ROWS ONLY;

```

- RFC7

```

WITH BuenClienteTiempo AS (
    SELECT DISTINCT r.usuarios_num_documento AS
num_documento_cliente
    FROM reservas r
    WHERE (r.fecha_salida - r.fecha_inicio) >= 14), BuenClienteConsumo AS (
    SELECT r.usuarios_num_documento AS num_documento_cliente,
SUM(c.costo) AS suma_consumos
    FROM reservas r
    JOIN consumos c ON r.id = c.habitaciones_id
    WHERE c.fecha_consumo >= (SELECT MAX(fecha_consumo) FROM
consumos) - INTERVAL '1' YEAR
    GROUP BY r.usuarios_num_documento
    HAVING SUM(c.costo) > 15000000)
SELECT u.num_documento AS num_documento_cliente, u.nombre AS
nombre_cliente,
CASE
    WHEN u.num_documento IN (SELECT num_documento_cliente FROM
BuenClienteConsumo) THEN TO_CHAR((SELECT suma_consumos FROM
BuenClienteConsumo WHERE num_documento_cliente = u.num_documento),
'$999,999,999.99')
    ELSE 'Buen usuario por tiempo'
END AS Razon
FROM usuarios u
WHERE u.num_documento IN (
    SELECT num_documento_cliente FROM BuenClienteConsumo
    UNION
    SELECT num_documento_cliente FROM BuenClienteTiempo);

```

- RFC8

```

WITH ConsumosPorSemana AS (
    SELECT t.tipo, TO_CHAR(c.fecha_consumo, 'IYYY-IW') AS semana,
COUNT(*) AS veces_consumido
    FROM consumos c INNER JOIN tiposservicio t ON c.tiposservicio_tipo =
t.tipo

```

```
WHERE c.fecha_consumo BETWEEN TO_DATE('2023-01-01', 'YYYY-MM-DD') AND TO_DATE('2023-12-31', 'YYYY-MM-DD')
```

```
GROUP BY t.tipo, TO_CHAR(c.fecha_consumo, 'YYYY-IW') )
```

```
SELECT s.tipo, s.semana, s.veces_consumido
```

```
FROM ConsumosPorSemana s
```

```
WHERE s.veces_consumido < 3;
```

- RFC9

```
SELECT
```

```
    r.usuarios_num_documento AS num_documento_cliente,
```

```
    r.usuarios_nombre AS nombre_cliente,
```

```
    c.tiposervicio_tipo AS tipo_servicio,
```

```
    COUNT(c.tiposervicio_tipo) AS veces_utilizado
```

```
FROM consumos c
```

```
LEFT JOIN reservan rv ON c.habitaciones_id = rv.habitaciones_id
```

```
LEFT JOIN reservas r ON rv.reservas_id = r.id
```

```
WHERE c.tiposervicio_tipo = 'spa' AND c.fecha_consumo BETWEEN  
TO_DATE('2023-01-01', 'YYYY-MM-DD') AND TO_DATE('2023-12-31',  
'YYYY-MM-DD')
```

```
GROUP BY r.usuarios_num_documento, r.usuarios_nombre,  
c.tiposervicio_tipo
```

```
ORDER BY r.usuarios_nombre ASC;
```

- RFC10

```
SELECT
```

```
    u.num_documento AS num_documento_cliente,
```

```
    u.nombre AS nombre_cliente
```

```
FROM
```

```
    usuarios u
```

```
WHERE
```

```

u.num_documento NOT IN (
    SELECT
        r.usuarios_num_documento AS num_documento_cliente
    FROM
        consumos c
    LEFT JOIN
        reservan rv ON c.habitaciones_id = rv.habitaciones_id
    LEFT JOIN
        reservas r ON rv.reservas_id = r.id
    WHERE
        c.tiposervicio_tipo = 'supermercado'
        AND c.fecha_consumo BETWEEN TO_DATE('2023-01-01', 'YYYY-MM-DD') AND TO_DATE('2023-12-31', 'YYYY-MM-DD')
    GROUP BY r.usuarios_num_documento, r.usuarios_nombre,
        c.tiposervicio_tipo
);

```

- RFC11

```

WITH Semanas AS (
    SELECT
        TO_CHAR(fecha_inicio, 'YYYY-IW') AS semana,
        reservan.reservas_id AS reserva_id,
        consumos.tiposervicio_tipo AS tipoServicio,
        reservan.habitaciones_id AS habitacion_id
    FROM reservas
    JOIN reservan ON reservan.reservas_id = reservas.id
    JOIN consumos ON reservan.habitaciones_id = consumos.habitaciones_id),
ServiciosConsumidos AS (
    SELECT

```

```

        semana,
        tipoServicio,
        COUNT(*) AS total_consumos
FROM Semanas
GROUP BY semana, tipoServicio),
HabitacionesSolicitadas AS (
    SELECT
        semana,
        habitacion_id,
        COUNT(*) AS total_solicitudes
    FROM Semanas
    GROUP BY semana, habitacion_id),
MaxMinServicios AS (
    SELECT
        semana,
        MAX(total_consumos) AS max_consumos,
        MIN(total_consumos) AS min_consumos
    FROM ServiciosConsumidos
    GROUP BY semana
), MaxMinHabitaciones AS (
    SELECT
        semana,
        MAX(total_solicitudes) AS max_solicitudes,
        MIN(total_solicitudes) AS min_consumos
    FROM HabitacionesSolicitadas
    GROUP BY semana)
SELECT
    s.semana,

```

```

MAX(CASE WHEN sc.total_consumos = mms.max_consumos THEN
tipoServicio END) AS servicio_mas_consumido,

MIN(CASE WHEN sc.total_consumos = mms.min_consumos THEN
tipoServicio END) AS servicio_menos_consumido,

MAX(CASE WHEN hs.total_solicitudes = mmh.max_solicitudes THEN
habitacion_id END) AS habitacion_mas_solicitada,

MIN(CASE WHEN hs.total_solicitudes = mmh.min_consumos THEN
habitacion_id END) AS habitacion_menos_solicitada

FROM (SELECT DISTINCT semana FROM Semanas) s

JOIN MaxMinServicios mms ON s.semana = mms.semana

JOIN MaxMinHabitaciones mmh ON s.semana = mmh.semana

LEFT JOIN ServiciosConsumidos sc ON s.semana = sc.semana

LEFT JOIN HabitacionesSolicitadas hs ON s.semana = hs.semana

GROUP BY s.semana

ORDER BY s.semana;

```

- RFC12

--- Reservas por trimestre

```

SELECT usuarios_nombre AS nombre_cliente, usuarios_num_documento AS
num_documento_cliente, usuarios_correo AS correo

FROM reservas

WHERE fecha_inicio BETWEEN TO_DATE('2023-01-01', 'YYYY-MM-DD')
AND TO_DATE('2023-03-31', 'YYYY-MM-DD')

AND usuarios_num_documento IN (

SELECT usuarios_num_documento

FROM reservas

WHERE fecha_inicio BETWEEN TO_DATE('2023-05-01', 'YYYY-MM-
DD') AND TO_DATE('2023-08-30', 'YYYY-MM-DD')

) AND usuarios_num_documento IN (

SELECT usuarios_num_documento

FROM reservas

```



```
WHERE fecha_inicio BETWEEN TO_DATE('2023-09-01', 'YYYY-MM-DD') AND TO_DATE('2023-12-30', 'YYYY-MM-DD');
```

---Por lo menos un consumo de servicio costoso

```
SELECT
```

```
    r.usuarios_nombre AS nombre_cliente,
```

```
    r.usuarios_num_documento AS num_documento_cliente,
```

```
    r.usuarios_correo AS correo
```

```
FROM reservas r
```

```
WHERE r.usuarios_num_documento IN (
```

```
    SELECT r2.usuarios_num_documento
```

```
    FROM reservas r2
```

```
    WHERE EXISTS (
```

```
        SELECT 1
```

```
        FROM consumos c
```

```
        WHERE c.fecha_consumo BETWEEN r2.fecha_inicio AND  
r2.fecha_salida AND c.costo >= 300000))
```

```
GROUP BY r.usuarios_num_documento, r.usuarios_nombre, r.usuarios_correo ;
```

```
SELECT
```

```
    r.usuarios_nombre AS nombre,
```

```
    r.usuarios_num_documento AS num_doc,
```

```
    r.usuarios_correo
```

```
FROM RESERVAS r
```

```
WHERE r.usuarios_num_documento NOT IN (
```

```
    SELECT DISTINCT re.usuarios_num_documento
```

```
    FROM
```

```
        RESERVASSERVICIO rs
```

```
    INNER JOIN RESERVAN ON rs.habitaciones_id = reservan.habitaciones_id
```

```
INNER JOIN RESERVAS re ON rs.dia BETWEEN re.fecha_inicio AND  
re.fecha_salida
```

```
WHERE rs.duracion_hora < 4)
```

```
GROUP BY r.usuarios_nombre, r.usuarios_num_documento, r.usuarios_correo;
```

- Índices:

Para mejorar el acceso a las tablas por medio de la agrupación de tipos de servicio:

```
CREATE  
INDEX idx_consumos_tiposervicio  
ON consumos(tiposervicio_tipo);
```

Para mejorar la consulta cuando tienen que acceder por la fecha a las reservas:

```
CREATE  
INDEX idx_consumos_fecha  
ON consumos(fecha_consumo);
```

Para la agrupación de reservas por usuario:

```
CREATE  
INDEX idx_reservas_num_doc  
ON reservas(usuarios_num_documento);
```

Para acceder a las reservas por habitación y fecha:

```
CREATE  
INDEX idx_consumos_habitaciones_fecha  
ON consumos(habitaciones_id, fecha_consumo);
```

Agrupación de reservas por el id de la habitación:

```
CREATE  
INDEX idx_reservan_habitaciones_id  
ON reservan(habitaciones_id);
```

- Comparación de planes de consulta:

Para RFC7:

Sin indicadores

SELECT STATEMENT			5	17
VIEW			1	2
Filter Predicates	NUM_DOCUMENTO_CLIENTE=:B1			
TABLE ACCESS	SYS.SYS_TEMP_0FD9D72F1_CF93870	FULL	1	2
VIEW			1	2
Filter Predicates	NUM_DOCUMENTO_CLIENTE=:B1			
TABLE ACCESS	SYS.SYS_TEMP_0FD9D72F1_CF93870	FULL	1	2
TEMP TABLE TRANSFORMATION				
LOAD AS SELECT	SYS_TEMP_0FD9D72F1_CF93870	(CURSOR DURATION MEMORY)		
HASH		GROUP BY	1	6
Filter Predicates	SUM(C.COSTO)>15000000			
NESTED LOOPS			1	4
NESTED LOOPS			1	4
TABLE ACCESS	CONSUMOS	FULL	1	4
Filter Predicates	C.FECHA_CONSUMO>= (SELECT MAX(FECHA_CONSUMO) FROM CONSUMOS CONSUMOS)-INTERVAL'+01-00' YEAR(2) TO MONTH			
SORT		AGGREGATE	1	
INDEX	CONSUMOS_PK	FULL SCAN	12	1
INDEX	RESERVAS_PK	UNIQUE SCAN	1	0
Access Predicates	R.ID=C.HABITACIONES_ID			
TABLE ACCESS	RESERVAS	BY INDEX ROWID	1	0
INDEX	USUARIOS_PK	FULL SCAN	1	1
Filter Predicates	EXISTS ((SELECT NUM_DOCUMENTO_CLIENTE FROM (SELECT /*+ CACHE (T1) */ C0 NUM_DOCUMENTO_CLIENTE,C1 SUMA_CONSUMOS FROM SYS.SYS_TEMP_0FD9D72F1_CF93870 T1) BUENCLIENTEC			
SORT		UNIQUE	2	6
UNION-ALL				
VIEW			1	2
Filter Predicates	NUM_DOCUMENTO_CLIENTE=:B1			
TABLE ACCESS	SYS.SYS_TEMP_0FD9D72F1_CF93870	FULL	1	2
TABLE ACCESS	RESERVAS	FULL	1	4
Filter Predicates	R.USUARIOS_NUM_DOCUMENTO=:B1			
AND	R.FECHA_SALIDA-R.FECHA_INICIO>=14			
			COST=4	

Con indicadores:

SELECT STATEMENT			5	13
VIEW			1	2
Filter Predicates	NUM_DOCUMENTO_CLIENTE=:B1			
TABLE ACCESS	SYS.SYS_TEMP_0FD9D7304_CF93870	FULL	1	2
VIEW			1	2
Filter Predicates	NUM_DOCUMENTO_CLIENTE=:B1			
TABLE ACCESS	SYS.SYS_TEMP_0FD9D7304_CF93870	FULL	1	2
TEMP TABLE TRANSFORMATION				
LOAD AS SELECT	SYS_TEMP_0FD9D7304_CF93870	(CURSOR DURATION MEMORY)		
HASH		GROUP BY	1	4
Filter Predicates	SUM(C.COSTO)>15000000			
NESTED LOOPS			1	2
NESTED LOOPS			1	2
TABLE ACCESS	CONSUMOS	BY INDEX ROWID BATCHED	1	2
INDEX	IDX_CONSUMOS_FECHA	RANGE SCAN	1	1
Access Predicates	C.FECHA_CONSUMO>= (SELECT MAX(FECHA_CONSUMO) FROM CONSUMOS CONSUMOS)-INTERVAL'+01-00' YEAR(2) TO MONTH			
SORT		AGGREGATE	1	
INDEX	CONSUMOS_PK	FULL SCAN	12	1
INDEX	RESERVAS_PK	UNIQUE SCAN	1	0
Access Predicates	R.ID=C.HABITACIONES_ID			
TABLE ACCESS	RESERVAS	BY INDEX ROWID	1	0
INDEX	USUARIOS_PK	FULL SCAN	1	1
Filter Predicates	EXISTS ((SELECT NUM_DOCUMENTO_CLIENTE FROM (SELECT /*+ CACHE (T1) */ C0 NUM_DOCUMENTO_CLIENTE,C1 SUMA_CONSUMOS FROM SYS.SYS_TEMP_0FD9D7304_CF93870 T1) BUENCLIENTEC			
SORT		UNIQUE	2	4
UNION-ALL			1	2
VIEW				
Filter Predicates	NUM_DOCUMENTO_CLIENTE=:B1			
TABLE ACCESS	SYS.SYS_TEMP_0FD9D7304_CF93870	FULL	1	2
TABLE ACCESS	RESERVAS	BY INDEX ROWID BATCHED	1	2
Filter Predicates	R.FECHA_SALIDA-R.FECHA_INICIO>=14			
INDEX	IDX_RESERVAS_NUM_DOC	RANGE SCAN	1	1
Access Predicates				

Para RFC9:

Sin indicadores

The diagram illustrates the execution plan for a complex SQL query. The plan is a tree structure showing the sequence of operations performed by the database engine. The operations are as follows:

- SELECT STATEMENT**: The root of the query.
- SORT**: A sort operation.
- HASH**: A hash operation.
- NESTED LOOPS**: A nested loops join operation.
- HASH JOIN**: A hash join operation.
- Access Predicates**: A predicate operation.
- NESTED LOOPS**: A nested loops join operation.
- STATISTICS COLLECTOR**: A statistics collector operation.
- INDEX**: An index operation.
- Access Predicates**: A predicate operation.
- AND**: An AND predicate operation.
- Filter Predicates**: A filter predicate operation.
- AND**: An AND predicate operation.
- INDEX**: An index operation.
- INDEX**: An index operation.
- Access Predicates**: A predicate operation.
- INDEX**: An index operation.
- TABLE ACCESS**: A table access operation.
- INDEX**: An index operation.
- Access Predicates**: A predicate operation.

The plan includes annotations for various predicates and join types, and a table on the right showing the cost and cardinality for each step.

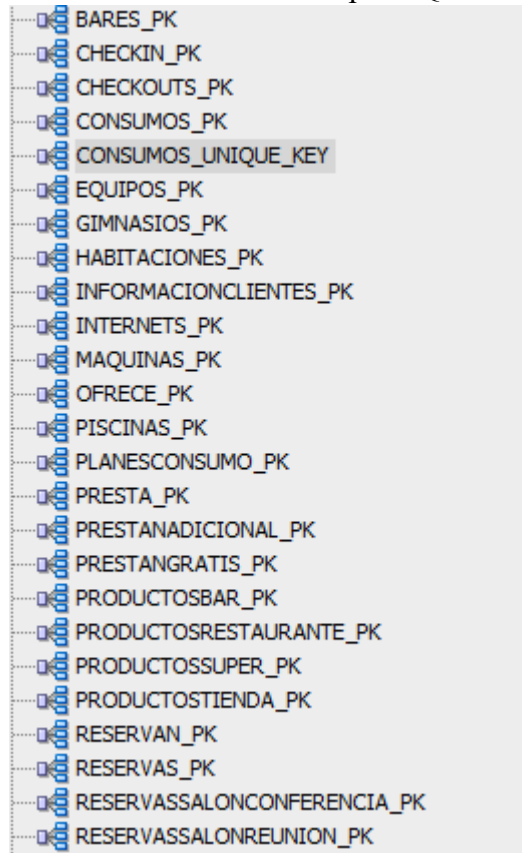
Operation	Cost	Cardinality
SELECT STATEMENT	6	4
SORT	6	4
HASH	6	4
NESTED LOOPS	6	2
HASH JOIN	6	2
Access Predicates	6	2
NESTED LOOPS	6	2
STATISTICS COLLECTOR	6	1
INDEX	6	1
Access Predicates	6	1
AND	6	1
Filter Predicates	6	1
AND	6	1
INDEX	6	1
INDEX	6	1
Access Predicates	6	1
INDEX	6	1
TABLE ACCESS	6	1
INDEX	6	1
Access Predicates	6	1

Operation	Access Predicates	Outer	Inner	Cost	Cardinality
SELECT STATEMENT				6	4
SORT				6	4
HASH				6	4
NESTED LOOPS				6	2
HASH JOIN				6	2
Access Predicates	C.HABITACIONES_ID=RV.HABITACIONES_ID(+)				
NESTED LOOPS		OUTER		6	2
STATISTICS COLLECTOR					
INDEX	CONSUMOS_PK	FULL SCAN		6	1
Access Predicates	AND (C.HABITACIONES_ID=RV.HABITACIONES_ID(+), C.CONSUMO_ID=RV.CONSUMO_ID(+))				
INDEX	RESERVAN_PK	RANGE SCAN		1	1
Access Predicates	C.HABITACIONES_ID=RV.HABITACIONES_ID(+)				
INDEX	RESERVAN_PK	FULL SCAN		5	1
INDEX	RESERVAS	BY INDEX ROWID		1	0
INDEX	RESERVAS_PK	UNIQUE SCAN		1	0
Access Predicates	RV.RESERVAS_ID=R.ID(+)				

Sin indicadores

OPERATION	TABLE	ACCESS	ROWS	BYTES	EST. COST	EST. ROWS	EST. BYTES
SELECT STATEMENT			1			22	
TEMP TABLE TRANSFORMATION							
LOAD AS SELECT							
HASH JOIN	SYS_TEMP_0FD9D72E9_CF93870	(CURSOR DURATION MEMORY)					
Access Predicates			12			2	
RESERVAN.HABITACIONES_ID=CONSUMOS.HABITACIONES_ID							
NESTED LOOPS			5			1	
NESTED LOOPS			5			1	
INDEX	RESERVAN_PK	FULL SCAN	5			1	
INDEX	RESERVAS_PK	UNIQUE SCAN	1			0	
Access Predicates							
RESERVAN.RESERVAS_ID=RESERVAS.ID							
TABLE ACCESS	RESERVAS	BY INDEX ROWID	1			0	
INDEX	CONSUMOS_PK	FULL SCAN	12			1	
LOAD AS SELECT	SYS_TEMP_0FD9D72EA_CF93870	(CURSOR DURATION MEMORY)					
HASH		GROUP BY	1			3	
VIEW	SYS_TEMP_0FD9D72E9_CF93870	FULL	12			2	
TABLE ACCESS	SYS_TEMP_0FD9D72EB_CF93870	(CURSOR DURATION MEMORY)	12			2	
LOAD AS SELECT							
HASH		GROUP BY	4			3	
VIEW	SYS_TEMP_0FD9D72E9_CF93870	FULL	12			2	
TABLE ACCESS	SYS_TEMP_0FD9D72E9_CF93870	FULL	12			2	
SORT		ORDER BY	1			14	

- Creados automáticamente por SQL:



Básicamente son las llaves primarias de cada tabla, importantes para reunir la información por un valor en específico.

- Para los casos de prueba se usaron pocos registros que cumplen las características para cada requerimiento, para cada uno se recibieron las respuestas esperadas.

RFC1:

Mostrar el dinero recolectado por servicios en cada habitación en el último año corrido.

Id de habitación	Costo total de servicios consumidos en el último año
1	320000
2	150487000
3	250000
4	5000000

RFC2

Rango de fechas: 01/01/2023, 30/11/2023

20 servicios más populares

Servicio	Cantidad consumos
spa	6
bar	3
restaurante	2
supermercado	1

RFC3

(muestra los índices redondeados)

Mostrar el índice de ocupacion de cada habitacion en el ultimo año.

Id de habitación	Días ocupados en el año	Indice de ocupacion en el año
2	8	2
4	8	2
5	91	25
3	8	2
6	0	0
1	0	0
7	0	0
8	0	0
9	0	0

RFC5

Rango de fechas: 01/01/2023, 30/11/2023. Num doc = 11111112

Total consumos

Num documento	Nombre	Total consumos
11111112	Susana Cardona	5000000

RFC6

Operacion Hotel Andes.

Reservas activas por fecha

Fecha	Reservas activas
2023-11-30 00:00:00.0	4
2023-11-29 00:00:00.0	4
2023-11-01 00:00:00.0	3
2023-07-29 00:00:00.0	2
2023-06-02 00:00:00.0	2
2023-12-02 00:00:00.0	2
2023-07-30 00:00:00.0	2
2023-12-01 00:00:00.0	2
2023-05-29 00:00:00.0	2
2023-01-02 00:00:00.0	1

Consumos por dia

Fecha	Total costo	Cantidad consumos
2023-11-02 00:00:00.0	150475000	4
2023-02-10 00:00:00.0	5000000	1
2023-01-10 00:00:00.0	200000	1
2023-10-01 00:00:00.0	150000	1
2023-11-04 00:00:00.0	78000	2
2023-11-03 00:00:00.0	54000	1
2023-08-10 00:00:00.0	50000	1
2023-09-10 00:00:00.0	50000	1

Dias con menos ocupacion

Fecha	Reservas activas
2023-01-02 00:00:00.0	1
2023-05-01 00:00:00.0	1
2023-12-11 00:00:00.0	1
2023-08-02 00:00:00.0	1
2023-01-01 00:00:00.0	1
2023-07-29 00:00:00.0	2
2023-12-02 00:00:00.0	2
2023-07-30 00:00:00.0	2
2023-05-29 00:00:00.0	2
2023-12-01 00:00:00.0	2

RFC7

Mostrar buenos clientes.

Muestra 'Buen usuario por tiempo' o el valor de consumos por el cual es un buen usuario

Num documento cliente	Nombre cliente	Razon
1765432	Eduardo benitez	\$150,807,000.00
111111112	Susana Cardona	Buen usuario por tiempo
333333333	Ricardo Ardila	Buen usuario por tiempo

RFC8

(Se probó con 3 consumos en una misma semana, no aparecen)

Consumos:

HABITACIONES_ID	TIPOSERVICIO_TIPO	DESCRIPCION	COSTO	FECHA_CONSUMO
1	1 spa	Eduardo Benitez se hizo un masaje	150000	01/10/23
2	3 spa	Consumo spa	50000	10/08/23
3	3 restaurante	Comida	200000	10/01/23
4	4 bar	Bebidas	5000000	10/02/23
5	1 bar	Bebidas	150000	02/11/23
6	2 restaurante	Pago comida para 100 personas	150150000	02/11/23
7	2 supermercado	Compras	125000	02/11/23
8	1 bar	Prueba2	20000	04/11/23
9	2 spa	RFC8	50000	02/11/23
10	2 spa	RFC8_2	54000	03/11/23
11	2 spa	RFC8_3	58000	04/11/23
12	2 spa	Consumo spa	50000	10/09/23

Resultado:

Mostrar los servicios en los que hubo menos de 3 consumos en 1 semana.

Tipo servicio	Numero semana	Veces consumido
supermercado	2023-44	1
bar	2023-44	2
spa	2023-36	1
restaurante	2023-44	1
spa	2023-32	1
bar	2023-06	1
restaurante	2023-02	1
spa	2023-39	1

RFC9

Rango de fechas: 01/01/2023, 30/11/2023. Tipo de servicio = 'spa'

Consumos por servicio

Num documento	Nombre	Tipo servicio	Total usado
1765432	Eduardo benitez	spa	5
333333333	Ricardo Ardila	spa	1

RFC10

Rango de fechas: 01/01/2023, 30/11/2023. Tipo de servicio = 'spa'

Consumos por servicio

Num documento	Nombre
204812	Doris Rodriguez
1653745	Armando Paredes
1786523	Leonardo Villegas
1928376	Adriana Macera
11111111	Jose Rodriguez
12333333	Gloria Trevi
12345678	Juan Perez
22222222	Jhon Doe
87654321	Tatiana Lopez
111111112	Susana Cardona

RFC11

Revisa las semanas en las que hubo entradas en reservas

Características por semana

Semana	Servicio mas consumido	Servicio menos consumido	Habitacion mas solicitada	Habitacion menos solicitada
2023-44	spa	bar	2	4
2023-48	bar	spa	1	1

RFC12

Clientes excelentes.

Por reservas en cada trimestre

Nombre	Num documento	correo
Eduardo benitez	1765432	e.benitez@gmail.com

Consumo +300,000

Nombre	Num documento	correo
Eduardo benitez	1765432	e.benitez@gmail.com
Ricardo Ardila	33333333	r.ardila@gmail.com
Susana Cardona	111111112	s.cardona@gmail.com

Reservas +4h

Nombre	Num documento	correo
Ricardo Ardila	33333333	r.ardila@gmail.com
Susana Cardona	111111112	s.cardona@gmail.com

- Datos

Se generaron los datos mediante la creación de un código en Python que selecciona valores aleatorios, también se hizo el uso de páginas web como generadores de sentencias de SQL para datos un poco más fáciles de crear, todo con el fin de generar el volumen de datos suficiente. El archivo de Python recibe como parámetro el numero de datos que quiere generar y este crea datos en .csv que después son transformados en sql para la facilidad de la carga masiva de datos, se escogieron nombres, apellidos y demás datos necesarios para generar el gran volumen de datos y la carga masiva de estos mismos.