

- **Nombre:** Salvador Moreno Sánchez
- **Centro asociado:** Motril (Granada)

Índice

- [Caso práctico “Bases de datos” - Informe](#)
- [Decisiones sobre el diseño](#)
- [Herramientas empleadas](#)

Caso práctico “Bases de datos” - Informe

Apartado 1: Crear la tabla Hoteles.

Código

```
CREATE TABLE Hoteles (
    CodHotel VARCHAR(10) NOT NULL,
    Nombre VARCHAR(30) NOT NULL,
    Direccion VARCHAR(50),
    Ciudad VARCHAR(30) NOT NULL,
    FecInaug DATE,
    Categoria NUMERIC(1) NOT NULL CHECK (Categoria BETWEEN 1 AND 5),
    NumHabitac NUMERIC(3) NOT NULL CHECK (NumHabitac > 0),
    PRIMARY KEY (CodHotel)
);
```

Apartado 2: Crear la tabla Clientes.

Código

```
CREATE TABLE Clientes (
    NIF VARCHAR(9) NOT NULL,
    Nombre VARCHAR(30) NOT NULL,
    Apellido1 VARCHAR(30) NOT NULL,
    Apellido2 VARCHAR(30),
    FecNacim DATE NOT NULL,
    Direccion VARCHAR(50),
    Ciudad VARCHAR(30) NOT NULL,
    PRIMARY KEY (NIF)
);
```

Apartado 3: Crear la tabla Reservas.

Código

```

CREATE TABLE Reservas (
    CodReserva VARCHAR(10) NOT NULL,
    CodHotel VARCHAR(10) NOT NULL,
    NIF VARCHAR(9) NOT NULL,
    FechaEntrada DATE,
    FechaReserva DATE NOT NULL,
    NumNoches NUMERIC(3) NOT NULL CHECK (NumNoches > 0),
    NumHabitac NUMERIC(3) NOT NULL CHECK (NumHabitac > 0),
    NumAdultos NUMERIC(4) NOT NULL DEFAULT 2 CHECK (NumAdultos > 0),
    NumNinyos NUMERIC(4) NOT NULL DEFAULT 0 CHECK (NumNinyos >= 0),
    PrecioNochHab NUMERIC(6,2) NOT NULL CHECK (PrecioNochHab > 0),
    PRIMARY KEY (CodReserva),
    FOREIGN KEY (CodHotel) REFERENCES Hoteles(CodHotel) ON DELETE CASCADE,
    FOREIGN KEY (NIF) REFERENCES Clientes(NIF) ON DELETE CASCADE
);

```

Poblar las tres tablas creadas con los datos proporcionados.

Resultado tabla Hoteles

	RFC codhotel	RBC nombre	RBC direccion	RBC ciudad	① fecinaug	123 categoria	123 numhabitac
1	H001	Hotel Sol	123 Calle Sol	Madrid	2000-05-20	4	100
2	H002	Hotel Luna	456 Calle Luna	Barcelona	2010-06-15	5	150
3	H003	Hotel Estrella	789 Calle Estrella	Valencia	2015-03-30	3	50
4	H004	Hotel Cometa	321 Calle Cometa	Sevilla	2020-01-22	2	30
5	H005	Hotel Galaxia	654 Calle Galaxia	Madrid	2005-12-05	4	80

Resultado tabla Clientes

	RFC nif	RBC nombre	RBC apellido1	RBC apellido2	① fecnacim	RBC direccion	RBC ciudad
1	12345678A	Ana	García	López	1985-04-15	111 Avda Prado	Madrid
2	87654321B	Luis	Martín	Santos	1972-08-25	222 Avda Olmos	Barcelona
3	11223344C	Carmen	Ruiz	Díaz	1990-01-10	333 Avda Mar	Valencia
4	44332211D	Jorge	Fernández	Moreno	1965-07-20	444 Avda Sierra	Sevilla
5	55667788E	Laura	Torres	Nieto	1979-12-30	555 Avda Sol	Madrid
6	23456789F	Sofía	Pérez	Gómez	1993-03-22	666 Avda Júpiter	Madrid
7	34567890G	Carlos	Jiménez	Fernández	1980-11-11	777 Avda Saturno	Barcelona
8	45678901H	Marta	López	Martínez	1976-02-05	888 Avda Neptuno	Valencia
9	56789012I	Antonio	García	Romero	1964-09-09	999 Avda Urano	Sevilla
10	67890123J	Beatriz	Vega	Prieto	1989-12-16	1010 Avda Plutón	Madrid

Resultado tabla Reservas

	RBC codreserva	RBC codhotel	RBC nif	fechaentrada	fechareserva	123 numnoches	123 numhabitac	123 numadultos	123 numninyos	123 precionochhab
8	R008	H002	44332211D	2023-06-05	2023-05-20	3	1	2	0	160
9	R009	H003	12345678A	2022-12-10	2022-11-25	4	3	2	2	85
10	R010	H004	87654321B	2022-11-15	2022-10-30	2	1	2	1	70
11	R011	H005	55667788E	2023-07-01	2023-06-15	6	2	2	3	95
12	R012	H001	11223344C	2023-08-20	2023-07-25	7	1	1	0	100
13	R013	H002	44332211D	2023-09-05	2023-08-01	10	2	2	2	180
14	R014	H003	12345678A	2022-07-25	2022-07-10	5	2	2	0	75
15	R015	H004	55667788E	2023-12-25	2023-12-01	14	1	2	2	65
16	R016	H005	44332211D	2023-10-10	2023-09-20	3	1	1	1	105
17	R017	H002	87654321B	2023-01-15	2022-12-20	8	3	3	0	140
18	R018	H001	11223344C	2023-11-01	2023-10-10	4	1	1	0	130
19	R019	H003	12345678A	2023-02-10	2023-01-15	3	1	2	1	90
20	R020	H004	44332211D	2023-05-20	2023-04-25	6	2	2	3	80
21	R021	H001	23456789F	2024-01-05	2023-12-20	4	1	2	1	100
22	R022	H002	34567890G	2024-02-15	2024-01-30	3	2	2	0	160
23	R023	H003	45678901H	2024-03-10	2024-02-20	7	1	1	2	85
24	R024	H004	56789012I	2024-04-20	2024-03-25	5	1	2	3	70
25	R025	H005	67890123J	2024-05-01	2024-04-15	6	2	2	2	95
26	R026	H001	23456789F	2024-06-25	2024-06-10	3	1	1	0	120
27	R027	H002	34567890G	2024-07-15	2024-06-30	8	1	2	1	150
28	R028	H003	45678901H	2024-08-05	2024-07-20	2	3	2	0	90
29	R029	H004	56789012I	2024-09-10	2024-08-25	10	2	2	2	80
30	R030	H005	67890123J	2024-10-20	2024-09-30	14	1	1	1	105
31	R031	H002	34567890G	2024-11-15	2024-10-31	5	2	3	0	140
32	R032	H001	23456789F	2024-12-10	2024-11-25	7	1	2	1	130
33	R033	H001	23456789F	2024-05-10	2024-04-14	5	1	2	1	110
34	R034	H002	34567890G	2024-06-15	2024-04-13	3	2	2	0	160
35	R035	H003	45678901H	2024-07-20	2024-04-12	7	1	1	2	85
36	R036	H004	56789012I	2024-08-05	2024-04-11	5	1	2	3	70
37	R037	H005	67890123J	2024-05-25	2024-04-10	6	2	2	2	95
38	R038	H001	23456789F	2024-06-30	2024-04-09	3	1	1	0	120
39	R039	H002	34567890G	2024-07-15	2024-04-08	8	1	2	1	150
40	R040	H003	45678901H	2024-08-25	2024-04-07	2	3	2	0	90
41	R041	H004	56789012I	2024-05-05	2024-04-06	10	2	2	2	80
42	R042	H005	67890123J	2024-06-20	2024-04-05	14	1	1	1	105
43	R043	H002	34567890G	2024-07-30	2024-04-04	5	2	3	0	140
44	R044	H001	23456789F	2024-08-10	2024-04-03	7	1	2	1	130

Apartado 4: CONSULTA Listar Hoteles de alta categoría.

Esta consulta selecciona todos los hoteles que tienen una categoría de 4 o 5, mostrando el CodHotel, nombre, ciudad y categoría.

Código

```
SELECT CodHotel, Nombre, Ciudad, Categoria
FROM Hoteles
WHERE Categoria >= 4;
```

Resultado

	RBC codhotel	RBC nombre	RBC ciudad	123 categoria
1	H001	Hotel Sol	Madrid	4
2	H002	Hotel Luna	Barcelona	5
3	H005	Hotel Galaxia	Madrid	4

Apartado 5: CONSULTA Contar Clientes por Ciudad.

Esta consulta cuenta cuántos clientes hay de cada ciudad, ordenados de forma descendente según ese número, lo cual es útil para entender la distribución geográfica de la clientela.

Código

```

SELECT Ciudad, COUNT(*) AS NumClientes
FROM Clientes
GROUP BY Ciudad
ORDER BY NumClientes DESC;

```

Resultado

	ABC ciudad	123 numclientes
1	Madrid	4
2	Sevilla	2
3	Barcelona	2
4	Valencia	2

Apartado 6: CONSULTA Detalle de las ocupaciones de los próximos 30 días.

Esta consulta recupera detalles de las reservas realizadas para entrar en los próximos 30 días, incluyendo el código de la reserva, el código del hotel, la fecha de reserva, la fecha de entrada, y el número de noches. El resultado ordenado por fecha de entrada ascendente.

Código

```

SELECT CodReserva, CodHotel, FechaReserva, FechaEntrada, NumNoches
FROM Reservas
WHERE FechaEntrada BETWEEN CURRENT_DATE AND CURRENT_DATE + INTERVAL '30' DAY
ORDER BY FechaEntrada ASC;

```

Resultado

	PP codreserva	RBC codhotel	⌚ fechareserva	⌚ fechaentrada	123 numnoches
1	R033	H001	2024-04-14	2024-05-10	5
2	R037	H005	2024-04-10	2024-05-25	6

Apartado 7: CONSULTA Ingresos futuros por hotel.

Calcular los ingresos futuros esperados de cada hotel basados en las reservas que aún no han comenzado, utilizando el número de habitaciones reservadas, número de noches y el precio por noche.

Código

```

SELECT R.CodHotel,
       H.Nombre AS NombreHotel,
       SUM(R.NumHabitac * R.NumNoches * R.PrecioNochHab) AS IngresosFuturos
  FROM Reservas R
 JOIN Hoteles H ON R.CodHotel = H.CodHotel
 WHERE R.FechaEntrada > CURRENT_DATE
 GROUP BY R.CodHotel, H.Nombre;

```

Resultado

	ABC codhotel ▾	ABC nombrehotel ▾	123 ingresosfuturos ▾
1	☒ H004	Hotel Cometa	1.950
2	☒ H005	Hotel Galaxia	4.080
3	☒ H002	Hotel Luna	6.160
4	☒ H001	Hotel Sol	3.090
5	☒ H003	Hotel Estrella	1.675

Apartado 8: CONSULTA Clientes frecuentes por ciudad (3 o más reservas).

Identificar los clientes que han realizado tres o más reservas en hoteles de una misma ciudad. Lista el nombre y apellido1 del cliente, la ciudad de los hoteles y el número total de reservas en hoteles de esa ciudad.

Código

```

SELECT C.Nombre,
       C.Apellido1,
       H.Ciudad,
       COUNT(*) AS NumReservas
  FROM Clientes C
 JOIN Reservas R ON C.NIF = R.NIF
 JOIN Hoteles H ON R.CodHotel = H.CodHotel
 GROUP BY C.NIF, H.Ciudad
 HAVING COUNT(*) >= 3;

```

Resultado

	ABC nombre	ABC apellido1	ABC ciudad	123 numreservas
1	Beatriz	Vega	Madrid	4
2	Carmen	Ruiz	Madrid	3
3	Marta	López	Valencia	4
4	Sofía	Pérez	Madrid	6
5	Ana	García	Valencia	3
6	Carlos	Jiménez	Barcelona	6
7	Antonio	García	Sevilla	4

Apartado 9: CONSULTA Reservas para entrar en un hotel en el mismo mes de su inauguración.

Listar todas las reservas cuya entrada coincide con el mes de aniversario de la inauguración del hotel, mostrando el nombre del hotel, fecha de inauguración, nombre del cliente y la fecha de entrada. El mes de la fecha de entrada tiene que coincidir con el mes de la fecha de inauguración.

Código

```
SELECT H.Nombre AS NombreHotel,
       H.FecInaug AS FechaInauguracion,
       C.Nombre AS NombreCliente,
       R.FechaEntrada
  FROM Reservas R
 JOIN Hoteles H ON R.CodHotel = H.CodHotel
 JOIN Clientes C ON R.NIF = C.NIF
 WHERE EXTRACT(MONTH FROM R.FechaEntrada) = EXTRACT(MONTH FROM H.FecInaug);
```

Resultado

	ABC nombrehotel	⌚ fechainauguracion	ABC nombreciente	⌚ fechaentrada
1	Hotel Luna	2010-06-15	Jorge	2023-06-05
2	Hotel Estrella	2015-03-30	Marta	2024-03-10
3	Hotel Sol	2000-05-20	Sofía	2024-05-10
4	Hotel Luna	2010-06-15	Carlos	2024-06-15

Comentarios

En un inicio, conformé la siguiente consulta, en la que empleaba la función `MONTH()` para extraer el mes de las fechas de inauguración y entrada:

```

SELECT H.Nombre AS NombreHotel,
       H.FecInaug AS FechaInauguracion,
       C.Nombre AS NombreCliente,
       R.FechaEntrada
  FROM Reservas R
 JOIN Hoteles H ON R.CodHotel = H.CodHotel
 JOIN Clientes C ON R.NIF = C.NIF
 WHERE MONTH(R.FechaEntrada) = MONTH(H.FecInaug);

```

Sin embargo, recibía un error que indicaba que la función `MONTH()` no estaba disponible en mi sistema de gestión de bases de datos. Investigando, llegué a la conclusión de que esto puede ocurrir si se está utilizando un sistema que no admite la función `MONTH()` o si la sintaxis específica del sistema es diferente.

Así, decidí buscar una alternativa: utilizar la función `EXTRACT()` para extraer partes específicas de una fecha, como el mes.

Apartado 10: CONSULTA Listado hoteles con meses de alta ocupación en 2023.

Listar los hoteles que tuvieron una ocupación de más del 2% en cualquier mes de 2023 mostrando nombre del hotel, año, mes y el porcentaje de ocupación. El porcentaje de ocupación en un mes se calcula como el número de habitaciones reservadas dividido por el número de habitaciones del hotel * 100. Se pide un dato tan pequeño de ocupación porque hay dadas de altas pocas reservas.

Código

```

SELECT H.Nombre AS NombreHotel,
       EXTRACT(YEAR FROM R.FechaEntrada) AS Año,
       EXTRACT(MONTH FROM R.FechaEntrada) AS Mes,
       (COUNT(*) * 100.0 / MAX(H.NumHabitac)) AS PorcentajeOcupacion
  FROM Reservas R
 JOIN Hoteles H ON R.CodHotel = H.CodHotel
 WHERE EXTRACT(YEAR FROM R.FechaEntrada) = 2023
 GROUP BY H.Nombre, EXTRACT(YEAR FROM R.FechaEntrada), EXTRACT(MONTH FROM R.FechaEntrada)
 HAVING (COUNT(*) * 100.0 / MAX(H.NumHabitac)) > 2;

```

Resultado

	ABC nombrehotel	123 año	123 mes	123 porcentajeocupacion
1	Hotel Cometa	2023	5	3,3333333333
2	Hotel Cometa	2023	12	3,3333333333

Comentarios

En un inicio, conformé la siguiente consulta:

```

SELECT H.Nombre AS NombreHotel,
       EXTRACT(YEAR FROM R.FechaEntrada) AS Año,
       EXTRACT(MONTH FROM R.FechaEntrada) AS Mes,
       (COUNT(*) * 100.0 / H.NumHabitac) AS PorcentajeOcupacion
  FROM Reservas R
 JOIN Hoteles H ON R.CodHotel = H.CodHotel
 WHERE EXTRACT(YEAR FROM R.FechaEntrada) = 2023
 GROUP BY H.Nombre, EXTRACT(YEAR FROM R.FechaEntrada), EXTRACT(MONTH FROM R.FechaEntrada)
 HAVING (COUNT(*) * 100.0 / H.NumHabitac) > 2;

```

Sin embargo, recibía el siguiente error:

```
ERROR: column "h.numhabitac" must appear in the GROUP BY clause or be used in an aggregate function
```

Este se debía a que estaba intentando utilizar la columna `H.NumHabitac` en la selección de la consulta sin agregarla a la cláusula `GROUP BY` o usarla en una función de agregación, siendo este segundo plantenamiento lo que debía hacer.

Apartado 11: CONSULTA Clientes sin reservas recientes.

Encontrar clientes que no hayan hecho ninguna reserva en el último año.

Código

```

SELECT NIF, Nombre, Apellido1, Apellido2, FecNacim, Direccion, Ciudad
  FROM Clientes
 WHERE NIF NOT IN (
    SELECT DISTINCT R.NIF
      FROM Reservas R
     WHERE R.FechaReserva >= CURRENT_DATE - INTERVAL '1 year'
);

```

Resultado

	nif	nombre	ap	Ctrl+click to open SQL console	fecnacim	direccion	ciudad
1	12345678A	Ana	García	López	1985-04-15	111 Avda Prado	Madrid
2	87654321B	Luis	Martín	Santos	1972-08-25	222 Avda Olmos	Barcelona

Apartado 12: CONSULTA Listado de hoteles y coste medio de habitación.

Listado de todos los hoteles con el nombre, categoría y precio medio de la habitación. Ordenar descendente por el precio medio.

Código

```

SELECT Nombre, Categoria, AVG(PrecioNochHab) AS PrecioMedioHabitacion
  FROM Hoteles
 JOIN Reservas ON Hoteles.CodHotel = Reservas.CodHotel
 GROUP BY Nombre, Categoria
 ORDER BY PrecioMedioHabitacion DESC;

```

Resultado

	ABC nombre	123 categoria	123 preciomediohabitacion
1	Hotel Luna	5	153
2	Hotel Sol	4	115,4545454545
3	Hotel Galaxia	4	101,4285714286
4	Hotel Estrella	3	86,25
5	Hotel Cometa	2	71,875

Apartado 13: VISTA VistaDetalleReservas con toda la información clave de las reservas.

Campos de la vista:

- **CodReserva:** El código identificador de la reserva.
- **HotelNombre:** El nombre del hotel asociado a la reserva.
- **HotelCiudad:** La ciudad donde está ubicado el hotel.
- **ClienteNombre:** El nombre del cliente que realizó la reserva.
- **ClienteApellido1:** El primer apellido del cliente.
- **FechaEntrada:** La fecha en la que el cliente tiene previsto iniciar su estancia.
- **NumNoches:** Número de noches que el cliente se hospedará en el hotel.
- **NumHabitac:** Número de habitaciones reservadas.
- **NumAdultos:** Número de adultos incluidos en la reserva.
- **NumNinyos:** Número de niños incluidos en la reserva.
- **PrecioNochHab:** Precio por noche y por habitación.

Código

```
CREATE VIEW VistaDetalleReservas AS
SELECT R.CodReserva,
       H.Nombre AS HotelNombre,
       H.Ciudad AS HotelCiudad,
       C.Nombre AS ClienteNombre,
       C.Apellido1 AS ClienteApellido1,
       R.FechaEntrada,
       R.NumNoches,
       R.NumHabitac,
       R.NumAdultos,
       R.NumNinyos,
       R.PrecioNochHab
FROM Reservas R
JOIN Hoteles H ON R.CodHotel = H.CodHotel
JOIN Clientes C ON R.NIF = C.NIF;
```

Resultado

	RBC codreserva	RBC hotelnombre	RBC hotelciudad	RBC clientenombre	RBC clienteapellido1	fechaentrada	123 numnoches	123 numhabitac	123 numadultos	123 numninyos	123 precionochhab
1	R001	Hotel Sol	Madrid	Ana	García	2023-04-01	3	2	2	1	100
2	R002	Hotel Luna	Barcelona	Luis	Martin	2023-04-15	5	1	2	2	150
3	R003	Hotel Estrella	Valencia	Carmen	Ruiz	2023-05-10	2	3	1	0	90
4	R004	Hotel Cometa	Sevilla	Jorge	Fernández	2022-12-25	7	1	2	3	60
5	R005	Hotel Sol	Madrid	Laura	Torres	2023-04-20	4	1	1	0	120
6	R006	Hotel Galaxia	Madrid	Ana	García	2023-04-25	3	1	2	2	110
7	R007	Hotel Sol	Madrid	Carmen	Ruiz	2023-03-15	5	2	1	1	110
8	R008	Hotel Luna	Barcelona	Jorge	Fernández	2023-06-05	3	1	2	0	160
9	R009	Hotel Estrella	Valencia	Ana	García	2022-12-10	4	3	2	2	85
10	R010	Hotel Cometa	Sevilla	Luis	Martin	2022-11-15	2	1	2	1	70
11	R011	Hotel Galaxia	Madrid	Laura	Torres	2023-07-01	6	2	2	3	95
12	R012	Hotel Sol	Madrid	Carmen	Ruiz	2023-08-20	7	1	1	0	100
13	R013	Hotel Luna	Barcelona	Jorge	Fernández	2023-09-05	10	2	2	2	180
14	R014	Hotel Estrella	Valencia	Ana	García	2022-07-25	5	2	2	0	75
15	R015	Hotel Cometa	Sevilla	Laura	Torres	2023-12-25	14	1	2	2	65
16	R016	Hotel Galaxia	Madrid	Jorge	Fernández	2023-10-10	3	1	1	1	105
17	R017	Hotel Luna	Barcelona	Luis	Martin	2023-01-15	8	3	3	0	140
18	R018	Hotel Sol	Madrid	Carmen	Ruiz	2023-11-01	4	1	1	0	130
19	R019	Hotel Estrella	Valencia	Ana	García	2023-02-10	3	1	2	1	90
20	R020	Hotel Cometa	Sevilla	Jorge	Fernández	2023-05-20	6	2	2	3	80
21	R021	Hotel Sol	Madrid	Sofía	Pérez	2024-01-05	4	1	2	1	100
22	R022	Hotel Luna	Barcelona	Carlos	Jiménez	2024-02-15	3	2	2	0	160
23	R023	Hotel Estrella	Valencia	Marta	López	2024-03-10	7	1	1	2	85
24	R024	Hotel Cometa	Sevilla	Antonio	García	2024-04-20	5	1	2	3	70
25	R025	Hotel Galaxia	Madrid	Beatriz	Vega	2024-05-01	6	2	2	2	95
26	R026	Hotel Sol	Madrid	Sofía	Pérez	2024-06-25	3	1	1	0	120
27	R027	Hotel Luna	Barcelona	Carlos	Jiménez	2024-07-15	8	1	2	1	150
28	R028	Hotel Estrella	Valencia	Marta	López	2024-08-05	2	3	2	0	90
29	R029	Hotel Cometa	Sevilla	Antonio	García	2024-09-10	10	2	2	2	80
30	R030	Hotel Galaxia	Madrid	Beatriz	Vega	2024-10-20	14	1	1	1	105
31	R031	Hotel Luna	Barcelona	Carlos	Jiménez	2024-11-15	5	2	3	0	140
32	R032	Hotel Sol	Madrid	Sofía	Pérez	2024-12-10	7	1	2	1	130
33	R033	Hotel Sol	Madrid	Sofía	Pérez	2024-05-10	5	1	2	1	110
34	R034	Hotel Luna	Barcelona	Carlos	Jiménez	2024-06-15	3	2	2	0	160
35	R035	Hotel Estrella	Valencia	Marta	López	2024-07-20	7	1	1	2	85
36	R036	Hotel Cometa	Sevilla	Antonio	García	2024-08-05	5	1	2	3	70
37	R037	Hotel Galaxia	Madrid	Beatriz	Vega	2024-05-25	6	2	2	2	95
38	R038	Hotel Sol	Madrid	Sofía	Pérez	2024-06-30	3	1	1	0	120

Apartado 14: CONSULTA Reservas realizadas en los últimos 30 días.

Utilizar la vista VistaDetalleReservas para realizar una consulta que encuentre todas las reservas con entrada en los próximos 30 días. Mostrar todos los campos de la vista.

Código

```
SELECT *
FROM VistaDetalleReservas
WHERE FechaEntrada BETWEEN CURRENT_DATE AND CURRENT_DATE + INTERVAL '30' DAY;
```

Resultado

	RBC codreserva	RBC hotelnombre	RBC hotelciudad	RBC clientenombre	RBC clienteapellido1	fechaentrada	123 numnoches	123 numhabitac	123 numadultos	123 numninyos	123 precionochhab
1	R033	Hotel Sol	Madrid	Sofía	Pérez	2024-05-10	5	1	2	1	110
2	R037	Hotel Galaxia	Madrid	Beatriz	Vega	2024-05-25	6	2	2	2	95

Comentarios

El enunciado es ambigüo, ya que pide los últimos y próximos 30 días. En mi caso, me he decantado por uno: los próximos 30 días.

Apartado 15: FUNCION CostoReserva() que calcule el costo de una reserva.

Crear la función CostoReserva() que tome como entrada el código de una reserva (CodReserva) y devuelva el costo de esa reserva específica. La función calculará el costo total de la reserva multiplicando el número de noches, el número de habitaciones y el precio por noche por habitación, todo a partir del código de reserva proporcionado.

Código

```
CREATE OR REPLACE FUNCTION CostoReserva(p_cod_reserva VARCHAR(10))
RETURNS NUMERIC AS
$$
DECLARE
    v_costo NUMERIC;
BEGIN
    SELECT (NumNoches * NumHabitac * PrecioNochHab)
    INTO v_costo
    FROM Reservas
    WHERE CodReserva = p_cod_reserva;

    RETURN v_costo;
END;
$$
LANGUAGE plpgsql;
```

Resultado

Llamada a la función `CostoReserva()` pasándole la reserva "R001" a modo de ejemplo

123 costoreserva	
1	600

Apartado 16: CONSULTA Costo medio de las reservas de cada cliente.

Empleando en la consulta la función `CostoReserva()` creada en el apartado anterior, listar el nombre de todos los clientes, el número de reservas que ha hecho, el costo medio de las reservas que ha hecho y el costo total de las reservas hechas.

Código

```
SELECT
    C.Nombre AS NombreCliente,
    COUNT(R.CodReserva) AS NumReservas,
    AVG(CostoReserva(R.CodReserva)) AS CostoMedioReservas,
    SUM(CostoReserva(R.CodReserva)) AS CostoTotalReservas
FROM Clientes C
LEFT JOIN Reservas R ON C.NIF = R.NIF
GROUP BY C.Nombre;
```

Resultado

	ABC nombrecliente	123 numreservas	123 costomedioreservas	123 costototalreservas
1	Carlos	6	1.186,6666666667	7.120
2	Marta	4	567,5	2.270
3	Sofía	6	581,6666666667	3.490
4	Ana	5	594	2.970
5	Jorge	5	1.155	5.775
6	Antonio	4	975	3.900
7	Carmen	4	715	2.860
8	Laura	3	843,3333333333	2.530
9	Beatriz	4	1.305	5.220
10	Luis	3	1.416,6666666667	4.250

Decisiones sobre el diseño

- Uso de Esquema Relacional:** el diseño del esquema relacional proporciona una estructura clara y organizada para almacenar y relacionar los datos. La división en entidades como Hoteles, Clientes y Reservas permite una gestión eficiente de la información relacionada con la cadena hotelera.
- Normalización de Datos:** se ha atendido a una adecuada normalización de datos en el esquema relacional, lo que minimiza la redundancia y mejora la integridad de los datos. Cada tabla representa una única entidad y se evita la duplicación de información.
- Utilización de Vistas:** la creación de la vista `VistaDetalleReservas` proporciona una capa de abstracción que simplifica las consultas complejas al encapsular la lógica de unión entre múltiples tablas. Esto promueve la modularidad y la reutilización del código.
- Definición de restricciones y tipos de datos:** la especificación de restricciones como claves primarias, claves foráneas y tipos de datos adecuados contribuye a garantizar la consistencia y la integridad de los datos almacenados en la base de datos.
- Uso de funciones y procedimientos:** la creación de la función `CostoReserva()` proporciona una forma eficiente de calcular el costo de una reserva específica. Esto promueve la reutilización del código y evita la duplicación de lógica en consultas repetitivas.
- Optimización de consultas:** en las consultas realizadas se evitan subconsultas innecesarias y se utilizan técnicas como `JOIN` y funciones de agregación de manera eficiente. Esto contribuye a mejorar el rendimiento de las consultas y a reducir la carga en el servidor de base de datos.
- Documentación y comentarios:** a modo de evolutivo, sería beneficioso incluir comentarios detallados en el código SQL para explicar la lógica detrás de las consultas, especialmente en consultas complejas o en la definición de la función `CostoReserva()`. Esto facilitaría la comprensión y el mantenimiento del código en el futuro.

En resumen, las decisiones de diseño tomadas a lo largo de este caso práctico reflejan buenas prácticas en el diseño y desarrollo de bases de datos relacionales, promoviendo la eficiencia, la integridad y la reutilización del código.

Herramientas empleadas

Cabe destacar que, en lugar de las herramientas recomendadas por el equipo docente (*Servidor MySQL* en local y *Cliente MySQL Workbench*), he utilizado **AWS** como instancia en la nube y **DBeaver** como aplicación de *software* cliente de SQL.

Como argumentación de mi elección, expongo lo siguiente:

1. **AWS:**
 - Escalabilidad y disponibilidad: AWS proporciona una infraestructura en la nube altamente escalable y disponible, lo que permite ajustar fácilmente los recursos de computación y almacenamiento según las necesidades del proyecto. Esto garantiza un rendimiento óptimo y una alta disponibilidad de los servicios de base de datos.
 - Seguridad y cumplimiento normativo: AWS ofrece una amplia gama de herramientas y servicios de seguridad

para proteger los datos sensibles y garantizar el cumplimiento normativo. Esto incluye opciones de cifrado de datos, control de acceso granular y auditorías de seguridad.

- Respaldo y recuperación de datos: *AWS* proporciona opciones integrales de respaldo y recuperación de datos, incluidas copias de seguridad automatizadas, almacenamiento redundante y capacidades de restauración granulares. Esto garantiza la integridad y la disponibilidad de los datos en caso de falla del sistema o pérdida de datos.

2. *DBeaver*:

- Versatilidad y compatibilidad: *DBeaver* es conocido por su versatilidad y compatibilidad con una amplia gama de sistemas de gestión de bases de datos (DBMS), incluidos MySQL, PostgreSQL, Oracle, SQL Server, entre otros. Esto permite a los desarrolladores trabajar con múltiples DBMS utilizando una única herramienta.
- Interfaz de usuario amigable: *DBeaver* ofrece una interfaz de usuario amigable e intuitiva, con funciones avanzadas de edición y visualización de datos. Esto facilita el desarrollo de bases de datos y la ejecución de consultas SQL.
- Soporte activo y comunidad: *DBeaver* cuenta con un sólido soporte técnico y una comunidad activa de usuarios y desarrolladores que proporcionan retroalimentación y contribuyen al desarrollo continuo de la herramienta.