

Práctica 4: Consultas, índices y vistas

Dado el script HotelBD.sql que contiene la base de datos de Hotel, realizar las siguientes consultas, vistas e índices:

1. Inserta el tipo de servicio Ocio.

```
INSERT INTO TIPOSERVICIO VALUES ('Ocio');
```

10 %

Messages

(1 row affected)

Completion time: 2024-04-18T21:43:20.4975227+01:00

2. Da de alta una reserva de la habitación 101 para el cliente 12345 para las noches del 2 al 4 de julio de 2024. El código de la reserva es autonumérico.

```
INSERT INTO RESERVAHABITAC (
    FECHAENTRADA,
    FECHASALIDA,
    IVA,
    NUMHABITACION,
    CLIENTE
) VALUES (
    '2024-07-02',
    '2024-07-04',
    0.07,
    101,
    '12345'
);
```

110 %

Messages

(1 row affected)

3. Actualiza el teléfono del cliente 12345. Su nuevo número es 123456789.

```
UPDATE CLIENTE
SET
  TELEFONO = '123456789'
WHERE
  IDCLIENTE = '12345';
```

110 %

Messages

(1 row affected)

4. Actualiza el precio de los servicios incrementándose en un 2%.

```
UPDATE SERVICIO
SET
  PRECIO = PRECIO * 1.02;
```

10 %

Messages

(3 rows affected)

5. Borra la reserva de la habitación 101 realizada anteriormente.

6. Borra los tipos de servicio que no tienen servicios definidos.

```
DELETE FROM TIPOSERVICIO
WHERE
    NOMBRESERVICIO NOT IN (
        SELECT
            NOMBRESERVICIO
        FROM
            SERVICIO
    );
```

0 %

Messages

(3 rows affected)

7. Crea una vista que devuelva los clientes cuyo apellido incluya la sílaba “le” ordenados por su identificador.

```
CREATE VIEW CLIENTESCON_LE AS
SELECT
    IDCLIENTE,
    PAIS,
    NOMBRE,
    PRIMERAPELLIDO,
    SEGUNDOAPELLIDO,
    DIRECCION,
    TELEFONO,
    OBSERVACIONES
FROM
    CLIENTE
WHERE
    PRIMERAPELLIDO LIKE '%le%'
    OR SEGUNDOAPELLIDO LIKE '%le%';
```

110 %

Messages

Commands completed successfully.

Completion time: 2024-04-19T11:45:37.3889182+01:00

8. Crea una vista que devuelva los clientes, ordenados por su primer apellido, que tengan alguna observación anotada.

```
CREATE VIEW ClientesConObservaciones AS
SELECT
    IDCLIENTE,
    PAIS,
    NOMBRE,
    PRIMERAPELLIDO,
    SEGUNDOAPELLIDO,
    DIRECCION,
    TELEFONO,
    OBSERVACIONES
FROM
    cliente
WHERE
    OBSERVACIONES IS NOT NULL;
```

110 %

Messages

Commands completed successfully.

Completion time: 2024-04-19T11:43:59.4345938+01:00

9. Crea una vista que devuelva los servicios cuyo precio supere los 5€ ordenados por su código de servicio.

```
CREATE VIEW serviciosPrecio_superior5 AS
SELECT
    *
FROM
    Servicio
WHERE
    PRECIO > 5;
```

10 %

Messages

Commands completed successfully.

Completion time: 2024-04-19T11:55:16.9937539+01:00

10. Crea una consulta que devuelva las habitaciones reservadas para el día 15 de marzo de 2023.

```
SELECT
*
FROM
ReservaHabitac
WHERE
fechaEntrada = '2023-03-15';
```

110 %

Results Messages

	idReserva	fechaEntrada	fechaSalida	iva	numHabitacion	cliente
1	1	2023-03-15	2023-03-25	0.07	101	12345
2	2	2023-03-15	2023-03-25	0.07	102	12345

11. Crea una consulta que devuelva los clientes procedentes de España y Francia.

```
SELECT
*
FROM
CLIENTE
WHERE
pais IN ('España', 'Francia');
```

110 %

Results Messages

	idCliente	pais	nombre	primerApellido	segundoApellido	direccion	telefono	observaciones
1	12345	España	Felipe	Iglesias	López	Avda Los Castros, 44	123456789	Buen cliente
2	44444	España	Cristina	García	García	Calle Mayor, 67	942456444	NULL
3	45678	Francia	Ludovic	Giuly	Bourquin	18 avenue Alsacen Cour	37890194	NULL

12. Crea una vista que devuelva los clientes que han utilizado el servicio de comedor.
13. Crea una vista que devuelva las características de cada habitación reservada.
14. Crea una consulta que devuelva los precios de los distintos tipos de habitación por temporada.

```
SELECT
    T.tipo,
    P.tipoHabitacion ,
    P.PRECIO
FROM
    PrecioHabitacion P
    INNER JOIN Temporada T
    ON P.temporada = T.temporada;
```

110 %

Results Messages

	tipo	tipoHabitacion	PRECIO
1	B	1	30,00
2	M	1	35,00
3	A	1	40,00
4	M	1	35,00
5	B	1	30,00
6	B	2	35,00
7	M	2	40,00
8	A	2	45,00
9	M	2	40,00
10	B	2	35,00
11	B	3	40,00
12	M	3	45,00
13	A	3	50,00
14	M	3	45,00
15	B	3	40,00
16	B	4	50,00

15. Crea una consulta que devuelva todos los clientes, y de aquellos que han realizado alguna reserva en marzo, indicar el no de reserva.

```
SELECT
    C.*,
    R.idReserva
FROM
    Cliente C
    LEFT JOIN ReservaHabitac R
    ON C.idCliente = R.cliente
    AND MONTH(R.fechaEntrada) = 3;
```

110 %

	idCliente	pais	nombre	primerApellido	segundoApellido	direccion	telefono	observaciones	idReserva
1	12345	España	Felipe	Iglesias	López	Avda Los Castros, 44	123456789	Buen cliente	1
2	12345	España	Felipe	Iglesias	López	Avda Los Castros, 44	123456789	Buen cliente	2
3	43215	Alemania	Hans	Schmidt	Müller	Hauptstrasse 123	65890234	Cliente exigente	NULL
4	44444	España	Cristina	García	García	Calle Mayor, 67	942456444	NULL	4
5	44444	España	Cristina	García	García	Calle Mayor, 67	942456444	NULL	5
6	44444	España	Cristina	García	García	Calle Mayor, 67	942456444	NULL	6
7	44444	España	Cristina	García	García	Calle Mayor, 67	942456444	NULL	7
8	45678	Francia	Ludovic	Giuly	Bourquin	18 avenue Alsacen Cour	37890194	NULL	NULL
9	45680	Italia	Giulia	Bianchi	NULL	Via Verdi 5	3209876543	NULL	NULL

16. Crea una vista con los servicios que nunca han sido contratados.

```
CREATE VIEW SERVICIOS_NO_CONTRATADOS AS
SELECT *
FROM SERVICIO
WHERE nombreServicio NOT IN (
    SELECT DISTINCT nombreServicio
    FROM ReservaHabitac
);
```

0 %

Messages

Commands completed successfully.

17. Crea una consulta que devuelva los clientes con el mismo primer apellido.
18. Crea una vista que devuelva el no de clientes por nacionalidad.

```
CREATE VIEW CLIENTES_POR_NACIONALIDAD AS
SELECT
    pais AS NACIONALIDAD,
    COUNT(*) AS NUMERO_DE_CLIENTES
FROM
    Cliente
GROUP BY
    pais;
```

110 %

Messages

Commands completed successfully.

Completion time: 2024-04-20T20:27:28.5024433+01:00

19. Crea una vista que devuelva el no de habitaciones por categoría de habitación.

```
CREATE VIEW HABITACIONES_POR_CATEGORIA AS
SELECT
    categoria AS CATEGORIA,
    COUNT(*) AS NUMERO_DE_HABITACIONES
FROM
    TipoHabitacion
GROUP BY
    categoria;
```

0 %

Messages

Commands completed successfully.

Completion time: 2024-04-20T20:29:46.9336406+01:00

20. Crea una vista que devuelva el no de servicios que se ofrecen por tipo de servicio.

```
CREATE VIEW SERVICIOS_POR_TIPO AS
SELECT
    nombreServicio AS TIPO_SERVICIO,
    COUNT(*) AS NUMERO_DE_SERVICIOS
FROM
    Servicio
GROUP BY
    nombreServicio;
```

110 %

Messages

Commands completed successfully.

21. Crea una consulta que devuelva el gasto en servicios realizado por cada reserva.

```
SELECT
    R.idReserva AS NUMERO_RESERVA,
    SUM(S.precio) AS GASTO_EN_SERVICIOS
FROM
    ReservaHabitac R
    INNER JOIN Gasto G ON R.idReserva = G.idReserva
    INNER JOIN Servicio S ON G.idServicio = S.idServicio
GROUP BY
    R.idReserva;
```

110 %

Results Messages

	NUMERO_RESERVA	GASTO_EN_SERVICIOS
1	1	20,40
2	4	2,04

22. Crea una consulta que devuelva el precio del servicio más caro y del más barato.

```
SELECT
    MAX(precio) AS PRECIO_SERVICIO_MAS_CARO,
    MIN(precio) AS PRECIO_SERVICIO_MAS_BARATO
FROM
    Servicio;
```

10 %

Results Messages

	PRECIO_SERVICIO_MAS_CARO	PRECIO_SERVICIO_MAS_BARATO
1	10,20	1,02

23. Crear una vista que muestre todos los datos del cliente excepto las observaciones. Actualizar el nombre de uno de los clientes utilizando esta vista.

```
CREATE VIEW CLIENTES_SIN_OBSERVACIONES AS
SELECT
    idCliente,
    NOMBRE,
    primerApellido,
    segundoApellido,
    pais
FROM
    CLIENTE;
```

%

Messages

Commands completed successfully.

Completion time: 2024-04-20T20:33:55.2187200+01:00

```
UPDATE Cliente
SET
    nombre = 'Nuevo Nombre'
WHERE
    idCliente = '1';
```

0 %

Messages

(0 rows affected)

24. Crear una vista que muestre el id, nombre y primer apellido de todos los clientes y el gasto total que han realizado en el hotel en sus diferentes estancias.

25. Crea un índice sobre el atributo pais de la tabla cliente, con objeto de agilizar las búsquedas.

```
CREATE INDEX IDX_PAIS ON CLIENTE (PAIS);
```

0 %

Messages

Commands completed successfully.

26. Crea un índice sobre el atributo numHabitacion de la tabla ReservaHabitac, con objeto de agilizar las búsquedas.

```
CREATE INDEX IDX_NUMHABITACION ON RESERVAHABITAC (NUMHABITACION);
```

110 %

Messages

Commands completed successfully.

27. Crea un índice sobre el atributo nombreServicio de la tabla Servicio, con objeto de agilizar las búsquedas.

```
CREATE INDEX IDX_NOMBRESERVICIO ON SERVICIO (NOMBRESERVICIO);
```

10 %

Messages

Commands completed successfully.

28. Añada una columna calculada en la tabla gastos que obtenga el importe por línea (cantidad y precio) sin aplicar el iva.

```
ALTER TABLE Gasto
ADD IMPORTE_SIN_IVA AS (cantidad * precio) PERSISTED;
```

110 %

Messages

Commands completed successfully.

29. Añada una columna en servicios que indique si el servicio se ofrece a los clientes o no.

```
ALTER TABLE Servicio
ADD SE_OFRECE_A_CLIENTES TINYINT;
```

0 %

Messages

Commands completed successfully.

30. Mostrar el precio de las habitaciones y sus respectivos tipos durante la temporada de verano (tipo 'A').

```
UPDATE SERVICIO
SET
  SE_OFRECE_A_CLIENTES = 1
WHERE
  nombreServicio = 'Servicio1';
```

0 %

Messages

(0 rows affected)