

SCRIPTS Y SALIDA

1. Consultas JOIN

```
1 -- INNER JOIN
2 SELECT reservacion.id_reservacion, cliente.nombre,
3 cliente.apellido, fecha_entrada, estado
4 FROM reservacion
5 INNER JOIN cliente ON reservacion.id_cliente = cliente.id_cliente;
```

	id_reservacion		nombre	apellido	fecha_entrada	estado
1	1		Juan	Pérez García	2024-03-15	confirmada
2	2		María	González López	2024-03-18	confirmada
3	3		John	Smith Brown	2024-03-20	completada
4	4		Emma	Wilson Taylor	2024-04-01	confirmada
5	5		Carlos	López Mendoza	2024-04-10	cancelada
6	6		Ana	Torres Vega	2024-04-20	confirmada
7	7		Luis	Rodríguez Castro	2024-05-01	confirmada
8	8		Sophie	Martin Bernard	2024-05-10	confirmada
9	9		Hans	Mueller Weber	2024-05-20	confirmada
10	10		Akiko	Tanaka Sato	2024-06-01	cancelada
11	11		Roberto	Silva Santos	2024-06-10	confirmada
12	12		Elena	Popov Ivanova	2024-06-20	confirmada
13	13		Mohammed	Al-Sayed	2024-07-01	confirmada
14	14		Isabella	Rossi Conti	2024-07-10	cancelada
15	15		Diego	Martínez Ruiz	2024-07-20	confirmada

```
1 -- LEFT JOIN
2 SELECT hotel.nombre, tipo_habitacion.nombre
3 FROM hotel
4 LEFT JOIN habitacion ON hotel.id_hotel = habitacion.id_hotel
5 LEFT JOIN tipo_habitacion ON habitacion.id_tipo = tipo_habitacion.id_tipo;
```

	nombre	nombre		
1	Marriott Lima Miraflores	Individual	17	InterContinental Lima (NULL)
2	Marriott Lima Miraflores	Doble	18	InterContinental Piura (NULL)
3	Marriott Lima Miraflores	Suite	19	Novotel Lima (NULL)
4	Marriott Cusco	Individual	20	Novotel Cusco (NULL)
5	Marriott Cusco	Doble	21	NH Lima (NULL)
6	Marriott Cusco	Suite	22	NH Trujillo (NULL)
7	Hilton Lima	Individual	23	Meliá Lima (NULL)
8	Hilton Lima	Doble	24	Meliá Arequipa (NULL)
9	Hilton Lima	Suite	25	Barceló Lima (NULL)
10	Hilton Arequipa	Individual	26	Barceló Iquitos (NULL)
11	Hilton Arequipa	Doble	27	RIU Plaza Lima (NULL)
12	Hilton Arequipa	Suite	28	RIU Cusco (NULL)
13	Hyatt Centric Lima	Individual	29	Mandarin Oriental Lima (NULL)
14	Hyatt Centric Lima	Doble	30	Mandarin Oriental Paracas (NULL)
15	Hyatt Centric Lima	Suite	31	Four Seasons Lima (NULL)
16	Hyatt Trujillo	(NULL)	32	Four Seasons Sacred Valley (NULL)

```

1 -- RIGHT JOIN
2 SELECT cadena_hotelera.nombre, hotel.nombre
3 FROM hotel
4 RIGHT JOIN cadena_hotelera ON hotel.id_cadena = cadena_hotelera.id_cadena;

```

ultado #1 (26r x 2c)

	nombre	nombre			
1	Marriott International	Marriott Lima Miraflores	14	Meliá Hotels International	Meliá Arequipa
2	Marriott International	Marriott Cusco	15	Barceló Hotel Group	Barceló Lima
3	Hilton Worldwide	Hilton Lima	16	Barceló Hotel Group	Barceló Iquitos
4	Hilton Worldwide	Hilton Arequipa	17	RIU Hotels & Resorts	RIU Plaza Lima
5	Hyatt Hotels Corporation	Hyatt Centric Lima	18	RIU Hotels & Resorts	RIU Cusco
6	Hyatt Hotels Corporation	Hyatt Trujillo	19	Mandarin Oriental Hotel Group	Mandarin Oriental Lima
7	InterContinental Hotels Group	InterContinental Lima	20	Mandarin Oriental Hotel Group	Mandarin Oriental Paracas
8	InterContinental Hotels Group	InterContinental Piura	21	Four Seasons Hotels	Four Seasons Lima
9	AccorHotels	Novotel Lima	22	Four Seasons Hotels	Four Seasons Sacred Valley
10	AccorHotels	Novotel Cusco	23	Rosewood Hotels	(NULL)
11	NH Hotel Group	NH Lima	24	Peninsula Hotels	(NULL)
12	NH Hotel Group	NH Trujillo	25	Shangri-La Hotels	(NULL)
			26	Kempinski Hotels	(NULL)

2. Consultas de DISTINCT

```

1 -- Hoteles por ciudad y cadena
2 SELECT DISTINCT ciudad, cadena_hotelera.nombre as cadena
3 FROM hotel
4 JOIN cadena_hotelera ON hotel.id_cadena = cadena_hotelera.id_cadena
5 ORDER BY ciudad;

```

ultado #1 (22r x 2c)

	ciudad	cadena			
1	Arequipa	Hilton Worldwide	12	Lima	NH Hotel Group
2	Arequipa	Meliá Hotels International	13	Lima	Meliá Hotels International
3	Cusco	Marriott International	14	Lima	Barceló Hotel Group
4	Cusco	AccorHotels	15	Lima	RIU Hotels & Resorts
5	Cusco	RIU Hotels & Resorts	16	Lima	Mandarin Oriental Hotel Group
6	Iquitos	Barceló Hotel Group	17	Lima	Four Seasons Hotels
7	Lima	Marriott International	18	Paracas	Mandarin Oriental Hotel Group
8	Lima	Hilton Worldwide	19	Piura	InterContinental Hotels Group
9	Lima	Hyatt Hotels Corporation	20	Trujillo	Hyatt Hotels Corporation
10	Lima	InterContinental Hotels Group	21	Trujillo	NH Hotel Group
11	Lima	AccorHotels	22	Urubamba	Four Seasons Hotels

```

1 -- Tipos de habitación por hotel
2 SELECT DISTINCT hotel.nombre as hotel, tipo_habitacion.nombre as tipo,
3 tipo_habitacion.capacidad
4 FROM hotel
5 JOIN habitacion ON hotel.id_hotel = habitacion.id_hotel
6 JOIN tipo_habitacion ON habitacion.id_tipo = tipo_habitacion.id_tipo
7 ORDER BY hotel.nombre;

```

resultado #1 (15r × 3c)

	hotel	tipo	capacidad
1	Hilton Arequipa	Individual	1
2	Hilton Arequipa	Doble	2
3	Hilton Arequipa	Suite	4
4	Hilton Lima	Individual	1
5	Hilton Lima	Doble	2
6	Hilton Lima	Suite	4
7	Hyatt Centric Lima	Individual	1
8	Hyatt Centric Lima	Doble	2
9	Hyatt Centric Lima	Suite	4
10	Marriott Cusco	Individual	1
11	Marriott Cusco	Doble	2
12	Marriott Cusco	Suite	4
13	Marriott Lima Miraflores	Individual	1
14	Marriott Lima Miraflores	Doble	2
15	Marriott Lima Miraflores	Suite	4

3. GROUP BY

```

1 -- GROUP BY
2 SELECT hotel.nombre, COUNT(habitacion.id_habitacion) as total_habitaciones
3 FROM hotel
4 JOIN habitacion ON hotel.id_hotel = habitacion.id_hotel
5 GROUP BY hotel.nombre;

```

hotel (5r × 2c)

	nombre	total_habitaciones
1	Marriott Lima Miraflores	3
2	Marriott Cusco	3
3	Hilton Lima	3
4	Hilton Arequipa	3
5	Hyatt Centric Lima	3

```

1 -- GROUP BY con HAVING
2 SELECT hotel.nombre, COUNT(reservacion.id_reservacion) as total_reservaciones
3 FROM hotel
4 JOIN habitacion ON hotel.id_hotel = habitacion.id_hotel
5 JOIN reservacion ON habitacion.id_habitacion = reservacion.id_habitacion
6 GROUP BY hotel.nombre
7 HAVING COUNT(reservacion.id_reservacion) > 2;

```

hotel (5r × 2c)		
	nombre	total_reservaciones
1	Marriott Lima Miraflores	3
2	Marriott Cusco	3
3	Hilton Lima	3
4	Hilton Arequipa	3
5	Hyatt Centric Lima	3

4. Procedimientos Almacenados

	Nombre	Tipo	Retorna
<input type="checkbox"/>	actualizar_estado_habitacion	PROCEDURE	Editar Ejecutar
<input type="checkbox"/>	cancelar_reservacion	PROCEDURE	Editar Ejecutar
<input type="checkbox"/>	nueva_reservacion	PROCEDURE	Editar Ejecutar

```

1  -- Procedimiento para registrar nueva reservaci n
2  DELIMITER //
3  CREATE PROCEDURE nueva_reservacion(
4      IN cliente_id INT,
5      IN habitacion_id INT,
6      IN fecha_in DATE,
7      IN fecha_out DATE,
8      IN num_huespedes INT,
9      IN metodo_pago VARCHAR(50)
10 )
11 BEGIN
12     INSERT INTO reservacion(id_cliente, id_habitacion,
13     fecha_entrada, fecha_salida, estado, numero_huespedes,
14     metodo_pago, fecha_reservacion)
15     VALUES (cliente_id, habitacion_id, fecha_in, fecha_out,
16     'confirmada', num_huespedes, metodo_pago, CURDATE());
17
18     UPDATE habitacion SET estado_actual = 'ocupada'
19     WHERE id_habitacion = habitacion_id;
20 END //
21 DELIMITER ;

```

```

77  -- Aqui vemos o verificamos si se guardo
78  SELECT id_habitacion, estado_actual
79  FROM habitacion WHERE id_habitacion = 5;
80

```

habitacion (1r × 2c)		
	id_habitacion	estado_actual
1	5	ocupada

```

1  -- Procedimiento para actualizar estado de habitaci n
2  DELIMITER //
3  CREATE PROCEDURE actualizar_estado_habitacion(
4      IN habitacion_id INT,
5      IN nuevo_estado ENUM('disponible','ocupada','mantenimiento')
6  )
7  BEGIN
8      UPDATE habitacion
9      SET estado_actual = nuevo_estado
10     WHERE id_habitacion = habitacion_id;
11 END //
12 DELIMITER ;

```

```

98 SELECT id_habitacion, estado_actual
99 FROM habitacion
100 WHERE id_habitacion = 3;
101

```

habitacion (1r × 2c)		
#	id_habitacion	estado_actual
1	3	ocupada

```

1  -- Procedimiento para cancelar reservaci n
2  DELIMITER //
3  CREATE PROCEDURE cancelar_reservacion(
4      IN reservacion_id INT
5  )
6  BEGIN
7      UPDATE reservacion
8      SET estado = 'cancelada'
9      WHERE id_reservacion = reservacion_id;
10
11     UPDATE habitacion
12     SET estado_actual = 'disponible'
13     WHERE id_habitacion = (SELECT id_habitacion
14                             FROM reservacion
15                             WHERE id_reservacion = reservacion_id);
16 END //
17 DELIMITER ;

```

```

24 -- cancelamos la reservacion con id_reser... = 5
25 CALL cancelar_reservacion(5);
26
27 -- Verificar el estado de la reservaci3n
28 SELECT id_reservacion, estado
29 FROM reservacion
30 WHERE id_reservacion = 5;
31

```

reservacion (1r × 2c)		
#	id_reservacion	estado
1	5	cancelada

Ejemplo de Ejecución:

Supongamos que hay una reservación con `id_reservacion = 5` y la habitación asociada es la habitación con `id_habitacion = 3`, esto sería para cancelar la reservación y liberar la habitación.

```
1 CALL cancelar_reservacion(5);
```

Luego pues vemos los cambios:

```
1 -- Verificar el estado de la reservacin
2 SELECT id_reservacion, estado
3 FROM reservacion
4 WHERE id_reservacion = 5;
```

Y nos saldra asi:

```
1 +-----+-----+
2 | id_reservacion | estado      |
3 +-----+-----+
4 | 5              | cancelada   |
5 +-----+-----+
```

Y para comprobar el estado de la habitación ejecutamos este script:

```
1 SELECT id_habitacion, estado_actual
2 FROM habitacion
3 WHERE id_habitacion = 3;
```

Y nos muestra esto:

```
1 +-----+-----+
2 | id_habitacion | estado_actual |
3 +-----+-----+
4 | 3              | disponible   |
5 +-----+-----+
```

Y pues aqui vemos que la `id_habitacion = 3` ahora está disponible después de cancelar la reservación.

5. Funciones Personalizadas

	Nombre	Tipo	Retor
<input type="checkbox"/>	actualizar_estado_habitacion	PROCEDURE	
<input type="checkbox"/>	cancelar_reservacion	PROCEDURE	
<input type="checkbox"/>	nueva_reservacion	PROCEDURE	

```
1 -- Funci n para calcular el precio total de una reservaci n
2 DELIMITER //
3 CREATE FUNCTION calcular_precio_total(dias INT, precio_base DECIMAL(10,2))
4 RETURNS DECIMAL(10,2)
5 DETERMINISTIC
6 BEGIN
7     RETURN dias * precio_base;
8 END //
9 DELIMITER ;
```

```
45 -- Diagmos que queremos calcular el precio de una
46 -- reservacion de 3 dias con uh precio de 100 diar
47 SELECT calcular_precio_total(3, 100);
48
49
```

resultado #1 (1r x 1c)	
	calcular_precio_total(3, 100)
1	300.0

```
1 -- Funci n para verificar disponibilidad de habitaci n
2 DELIMITER //
3 CREATE FUNCTION esta_disponible(habitacion_id INT)
4 RETURNS BOOLEAN
5 DETERMINISTIC
6 BEGIN
7     DECLARE estado VARCHAR(20);
8     SELECT estado_actual INTO estado FROM habitacion
9     WHERE id_habitacion = habitacion_id;
10    RETURN estado = 'disponible';
11 END //
12 DELIMITER ;
```

```
3 -- esto es para verificar si la habitacion
4 -- con id 3 esta dispo.
5 SELECT esta_disponible(3);
6 -- En este caso sale 0 sig. que no esta
7 -- disponible y 1 si estaria pero no es el cas
8
```

sultado #1 (1r x 1c)	
	esta_disponible(3)
1	0

```
1 -- Funci n para contar reservaciones activas de un cliente
2 DELIMITER //
3 CREATE FUNCTION contar_reservaciones_activas(cliente_id INT)
4 RETURNS INT
5 DETERMINISTIC
6 BEGIN
7     DECLARE total INT;
8     SELECT COUNT(*) INTO total
9     FROM reservacion
10    WHERE id_cliente = cliente_id
11    AND estado = 'confirmada';
12    RETURN total;
13 END //
14 DELIMITER ;
```

```

5  -- si queremos ver las reservaciones
6  -- activas del cliente con id 5
7  SELECT contar_reservaciones_activas(5);

```

Resultado #1 (1r x 1c)

contar_reservaciones_activas(5)	
1	0

6. Triggers

```

1  -- Trigger para actualizar estado de habitación después de una reservación
2  DELIMITER //
3  CREATE TRIGGER actualizar_estado_despues_reserva
4  AFTER INSERT ON reservacion
5  FOR EACH ROW
6  BEGIN
7      UPDATE habitacion
8      SET estado_actual = 'ocupada'
9      WHERE id_habitacion = NEW.id_habitacion;
10 END //
11 DELIMITER ;

```

```

33 -- Para crear una reservación
34 DELIMITER //
35 CREATE PROCEDURE crear_reservacion(
36     cliente_id INT,
37     habitacion_id INT,
38     fecha_inicio DATE,
39     fecha_fin DATE,
40     huespedes INT
41 )
42 BEGIN
43     INSERT INTO reservacion (id_cliente,
44                             id_habitacion, fecha_inicio,
45                             fecha_fin, estado, numero_huespedes)
46     VALUES (cliente_id, habitacion_id,
47             fecha_inicio, fecha_fin, 'confirmada', huespedes);
48 END //
49 DELIMITER ;
50
51 CALL crear_reservacion(3, 2, '2024-12-01', '2024-12-05', 2);

```

3	2	NULL	NULL	confirmada	2	NULL	NULL	2024-12-01	2024-12-05
3	2	NULL	NULL	confirmada	2	NULL	NULL	2024-12-01	2024-12-05

```

1  -- Trigger para verificar capacidad antes de insertar reservación
2  DELIMITER //
3  CREATE TRIGGER verificar_capacidad_antes_reserva
4  BEFORE INSERT ON reservacion
5  FOR EACH ROW
6  BEGIN
7      DECLARE max_capacidad INT;
8      SELECT capacidad INTO max_capacidad
9      FROM habitacion h
10     JOIN tipo_habitacion t ON h.id_tipo = t.id_tipo
11     WHERE h.id_habitacion = NEW.id_habitacion;
12
13     IF NEW.numero_huespedes > max_capacidad THEN
14         SIGNAL SQLSTATE '45000'
15         SET MESSAGE_TEXT = 'Número de huéspedes excede la capacidad de la habitación';

```



```

16     END IF;
17 END //
18 DELIMITER ;

```

```

-- ejemplo
INSERT INTO reservacion (id_cliente, id_habitacion, fecha_entrada, fecha_salida, numero_huespedes)
VALUES (1, 3, '2024-12-01', '2024-12-05', 3);

SELECT * FROM reservacion;

```

reservacion (19r × 11c)								
id_reservacion	id_cliente	id_habitacion	fecha_entrada	fecha_salida	estado	numero_huespedes	metodo_pago	
20	1	3	2024-12-01	2024-12-05	(NULL)	3	(NULL)	

```

1 -- Trigger para liberar habitacion despues de cancelar reservacion
2 DELIMITER //
3 CREATE TRIGGER liberar_habitacion_despues_cancelar
4 AFTER UPDATE ON reservacion
5 FOR EACH ROW
6 BEGIN
7     IF NEW.estado = 'cancelada' AND OLD.estado = 'confirmada' THEN
8         UPDATE habitacion
9         SET estado_actual = 'disponible'
10        WHERE id_habitacion = NEW.id_habitacion;
11    END IF;
12 END //
13 DELIMITER ;

```

```

77 -- ejemplo
78 UPDATE reservacion
79 SET estado = 'cancelada'
80 WHERE id_reservacion = 5;
81
82 SELECT estado_actual
83 FROM habitacion
84 WHERE id_habitacion = 3;
85
86

```

habitacion (1r × 1c)	
	estado_actual
1	ocupada