## Rooms

(room\_id, room\_number, room\_category, capacity, room\_price\_per\_day)

room\_status(0, 1), room\_state(dirty, broken, clean)

## Customers

(customer\_id, forename, surname, nif, phone, email) pendiente guests

customer\_status(0, 1)

## Reservations

(reservation\_id, customer\_id, room\_id, date\_in, date\_out, reservation\_price\_per\_day)

reservation\_state(booked, cancelled, checked\_in, checked\_out)

## Crear selector de habitaciones disponibles

DELIMITER $$

CREATE PROCEDURE search\_available\_rooms(

IN var\_date\_in DATE,

IN var\_date\_out DATE

)

BEGIN

-- Selecciona las habitaciones disponibles en las fechas proporcionadas

SELECT room\_id

FROM rooms

WHERE room\_id NOT IN (

SELECT room\_id

FROM reservations

WHERE NOT (reservations.date\_out <= var\_date\_in OR reservations.date\_in >= var\_date\_out)

);

END $$

DELIMITER ;

## VIEW reservations

CREATE VIEW reservations\_view AS

SELECT reservations.reservation\_id, reservations.customer\_id, customers.forename, customers.surname, customers.nif, customers.phone, reservations.room\_id, rooms.room\_number, reservations.date\_in, reservations.date\_out, reservations.price\_per\_day, reservations.status, DATEDIFF(reservations.date\_out, reservations.date\_in) AS total\_days, reservations.price\_per\_day \* DATEDIFF(reservations.date\_out, reservations.date\_in) AS total\_price

FROM reservations

INNER JOIN rooms

ON reservations.room\_id = rooms.room\_id

INNER JOIN customers

ON reservations.customer\_id = customers.customer\_id;

## CREAR room\_category

CREATE TABLE room\_category(room\_category\_id INT, room\_category\_name VARCHAR(50), room\_category\_price DECIMAL);

## VIEW reservations(aplicando room\_category)

CREATE VIEW `070\_reservations\_view` AS

SELECT

`r`.`reservation\_id` AS `reservation\_id`,

`r`.`user\_id` AS `user\_user\_id`,

`u`.`user\_forename` AS `user\_forename`,

`u`.`user\_surname` AS `user\_surname`,

`u`.`user\_nif` AS `user\_nif`,

`u`.`user\_email` AS `user\_email`,

u.user\_password AS `user\_password`,

`u`.`user\_phone` AS `user\_phone`,

`u`.`user\_birthday` AS `user\_birthday`,

u.user\_status AS `user\_status`,

u.customer\_id AS `customer\_id`,

`u`.`customer\_requirements` AS `customer\_requirements`,

`r`.`room\_id` AS `room\_id`,

`rv`.`room\_number` AS `room\_number`,

`rv`.`room\_category\_name` AS `room\_category\_name`,

`r`.`date\_in` AS `date\_in`,

`r`.`date\_out` AS `date\_out`,

`r`.`price\_per\_day` AS `price\_per\_day`,

`r`.`status` AS `status`,

TO\_DAYS(`r`.`date\_out`) - TO\_DAYS(`r`.`date\_in`) AS `total\_days`,

`r`.`price\_per\_day` \*(

TO\_DAYS(`r`.`date\_out`) - TO\_DAYS(`r`.`date\_in`)

) AS `total\_price`

FROM

`070\_reservations` AS r

LEFT JOIN `070\_rooms\_view` AS rv

ON r.room\_id = rv.room\_id

LEFT JOIN `070\_users\_view` AS u

ON r.user\_id = u.user\_id;

## VIEW users

CREATE VIEW `070\_users\_view` AS

SELECT u.user\_id, u.user\_forename, u.user\_surname, u.user\_nif, u.user\_email, u.user\_password, u.user\_phone, u.user\_birthday,

c.customer\_id, c.customer\_requirements, e.employee\_id, e.date\_hiring, e.position\_id, e.position\_name, e.position\_salary

FROM `070\_users` AS u

LEFT JOIN `070\_customers` AS c ON u.customer\_id = c.customer\_id

LEFT JOIN `070\_employees\_view` AS e ON u.employee\_id = e.employee\_id;

## PROCEDURE seleccionar categorias

DELIMITER $$

CREATE PROCEDURE `070\_select\_category\_name\_per\_price`()

BEGIN

SELECT room\_category\_name

FROM `070\_room\_category`

GROUP BY room\_category\_id

ORDER BY room\_category\_price\_per\_day;

END $$

DELIMITER ;

## CREATE TABLE clients

CREATE TABLE clients (

client\_id INT AUTO\_INCREMENT PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

second\_name VARCHAR(50) NOT NULL,

dni VARCHAR(20) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

phone VARCHAR(15),

password VARCHAR(50) NOT NULL,

birthday DATE

);

## CREATE TABLE rooms

CREATE TABLE rooms (

room\_id INT AUTO\_INCREMENT PRIMARY KEY,

room\_number VARCHAR(10) NOT NULL,

room\_type ENUM('Simple', 'Double', 'Suite') DEFAULT 'Simple',

price\_per\_night DECIMAL(10, 2) NOT NULL,

description VARCHAR(200),

status ENUM('Available', 'Occupied', 'Maintenance') DEFAULT 'Available',

max\_occupancy INT NOT NULL

);

## CREATE TABLE reservations

CREATE TABLE reservations (

reservation\_id INT AUTO\_INCREMENT PRIMARY KEY,

client\_id INT,

room\_id INT,

check\_in\_date DATE NOT NULL,

check\_out\_date DATE NOT NULL,

total\_amount DECIMAL(10, 2) NOT NULL,

status ENUM('confirmed', 'cancelled', 'pending') DEFAULT 'pending',

CONSTRAINT fk\_client FOREIGN KEY (client\_id) REFERENCES clients(client\_id),

CONSTRAINT fk\_room FOREIGN KEY (room\_id) REFERENCES rooms(room\_id)

);

## CREATE TABLE payments

CREATE TABLE payments (

payment\_id INT AUTO\_INCREMENT PRIMARY KEY,

reservation\_id INT,

payment\_date DATETIME NOT NULL,

amount DECIMAL(10, 2) NOT NULL,

payment\_method ENUM('credit\_card', 'cash', 'paypal') NOT NULL,

payment\_status ENUM('completed', 'failed', 'pending') DEFAULT 'pending',

CONSTRAINT fk\_reservation FOREIGN KEY (reservation\_id) REFERENCES reservations(reservation\_id)

);