Hotel database by Temirlan Abeu

Here is my tables:

The **guest** table likely stores information about your hotel guests, such as their name, contact details, and any notes or special requests they may have.

CREATE TABLE guests (  
 guest\_id INTEGER PRIMARY KEY,  
 first\_name TEXT,  
 last\_name TEXT,  
 email TEXT,  
 phone\_number TEXT,  
 special\_requests TEXT  
);

The **booking** table stores information about bookings made by guests, including the dates of their stay, the type of room they have booked, and any other services or amenities they have requested.

CREATE TABLE bookings (  
 booking\_id INTEGER PRIMARY KEY,  
 guest\_id INTEGER,  
 room\_id INTEGER,  
 check\_in\_date DATE,  
 check\_out\_date DATE,  
 FOREIGN KEY (guest\_id) REFERENCES guests(guest\_id),  
 FOREIGN KEY (room\_id) REFERENCES rooms(room\_id)  
);

The **payment** table stores information about payments made by guests, including the amount paid, the payment method, and any relevant transaction details.

CREATE TABLE payments (  
 payment\_id INTEGER PRIMARY KEY,  
 bill\_id INTEGER,  
 payment\_method TEXT,  
 amount REAL,  
 FOREIGN KEY (bill\_id) REFERENCES bills(bill\_id)  
);

The **bills** table stores information about bills generated for guests during their stay, including the items or services that were charged, the amount due, and any relevant notes or details.

CREATE TABLE bills (  
 bill\_id INTEGER PRIMARY KEY,  
 guest\_id INTEGER,  
 room\_id INTEGER,  
 service\_id INTEGER,  
 date DATE,  
 amount REAL,  
 FOREIGN KEY (guest\_id) REFERENCES guests(guest\_id),  
 FOREIGN KEY (room\_id) REFERENCES rooms(room\_id),  
 FOREIGN KEY (service\_id) REFERENCES services(service\_id)  
);

The **departments** table stores information about the various departments in your hotel, such as housekeeping, front desk, and maintenance.

CREATE TABLE departments (  
 department\_id INTEGER PRIMARY KEY,  
 name TEXT,  
 description TEXT  
);

The **employees** table stores information about your hotel employees, including their names, contact details, job titles, and any relevant notes or details.

CREATE TABLE employees (  
 employee\_id INTEGER PRIMARY KEY,  
 first\_name TEXT,  
 last\_name TEXT,  
 email TEXT,  
 phone\_number TEXT,  
 job\_title TEXT  
);

The **reviews** table stores guest reviews of your hotel, including the guest's name, the date of their stay, and their comments or feedback.

CREATE TABLE reviews (  
 review\_id INTEGER PRIMARY KEY,  
 guest\_id INTEGER,  
 rating INTEGER,  
 review TEXT,  
 FOREIGN KEY (guest\_id) REFERENCES guests(guest\_id)  
);

The **room\_categories** table stores information about the different categories of rooms available in your hotel, such as standard rooms, suites, and executive rooms.

CREATE TABLE room\_categories (  
 category\_id INTEGER PRIMARY KEY,  
 name TEXT,  
 description TEXT  
);

The **rooms** table stores information about the individual rooms in your hotel, including the room number, category, and any relevant notes or details.

CREATE TABLE rooms (  
 room\_id INTEGER PRIMARY KEY,  
 room\_number TEXT,  
 category\_id INTEGER,  
 FOREIGN KEY (category\_id) REFERENCES room\_categories(category\_id)  
);

The **service** table stores information about the various services offered by your hotel, such as room service, laundry, and spa services.

CREATE TABLE services (  
 service\_id INTEGER PRIMARY KEY,  
 name TEXT,  
 description TEXT,  
 price REAL  
);

The **service\_bookings** table stores information about bookings made for hotel services, including the service requested, the dates it was booked for, and any relevant notes or details.

CREATE TABLE service\_bookings (  
 service\_booking\_id INTEGER PRIMARY KEY,  
 service\_id INTEGER,  
 guest\_id INTEGER,  
 date DATE,  
 FOREIGN KEY (service\_id) REFERENCES services(service\_id),  
 FOREIGN KEY (guest\_id) REFERENCES guests(guest\_id)  
);

Here are my queries:

This part adds column price into room categories table

ALTER TABLE room\_categories ADD COLUMN price REAL;

This part adds column which contains employees number on department table

ALTER TABLE departments ADD COLUMN num\_employees INTEGER;

This trigger sends email reply to guest who wrote review which can contain guests name, which is taken from table’s data

CREATE TRIGGER send\_review\_notification AFTER INSERT ON reviews  
FOR EACH ROW BEGIN  
 SELECT send\_email(  
 "hotel.management@example.com",  
 "New guest review",  
 CONCAT("A new review has been left by ", (SELECT first\_name FROM guests WHERE guest\_id = NEW.guest\_id), " ", (SELECT last\_name FROM guests WHERE guest\_id = NEW.guest\_id), ": ", NEW.review)  
 );  
END;  
This part adds column total amount due, then I wrote trigger to get all amount that guest spent in specified service  
alter table service\_bookings  
add column total\_amount\_due;  
  
CREATE TRIGGER update\_total\_amount\_due AFTER INSERT ON service\_bookings  
FOR EACH ROW BEGIN  
 UPDATE bookings  
 SET total\_amount\_due = (SELECT *SUM*(amount) FROM bills WHERE guest\_id = NEW.guest\_id AND date BETWEEN check\_in\_date AND check\_out\_date)  
 WHERE guest\_id = NEW.guest\_id;  
END;  
This part displays how much in average guests spent in our hotel  
SELECT *AVG*(*strftime*(check\_out\_date - check\_in\_date)) as Average\_nights FROM bookings;  
This query displays the name of the room category and the number of bookings for each category, grouped by the room category name, and ordered by the number of bookings in descending order. The query is then limiting the output to just the top row, which will be the room category with the most bookings.  
SELECT rc.name, *COUNT*(r.room\_id) AS num\_bookings  
FROM rooms r  
INNER JOIN room\_categories rc ON r.category\_id = rc.category\_id  
GROUP BY rc.name  
ORDER BY num\_bookings DESC  
LIMIT 1;  
This query is retrieving the total sum of the amount column from the bills table for rows where the service\_id column is null, and returning the result as a column called total\_revenue. This would give you the total revenue generated from bills that do not have a corresponding service associated with them.  
SELECT *SUM*(b.amount) AS total\_revenue  
FROM bills b  
WHERE b.service\_id IS NULL;  
This query is selecting the first and last names of guests who have booked both a standard and deluxe room.  
SELECT g.first\_name, g.last\_name, r.room\_number  
FROM guests g  
INNER JOIN bookings b ON g.guest\_id = b.guest\_id  
INNER JOIN rooms r ON b.room\_id = r.room\_id  
WHERE r.category\_id IN (SELECT category\_id FROM room\_categories WHERE name IN ("Standard", "Deluxe"))  
GROUP BY g.guest\_id  
HAVING *COUNT*(DISTINCT r.category\_id) = 2;

This SQL query is selecting the first and last names of guests who have only booked standard rooms.

SELECT g.first\_name, g.last\_name, r.room\_number  
FROM guests g  
INNER JOIN bookings b ON g.guest\_id = b.guest\_id  
INNER JOIN rooms r ON b.room\_id = r.room\_id  
WHERE r.category\_id IN (SELECT category\_id FROM room\_categories WHERE name = "Standard")  
AND NOT *EXISTS* (  
 SELECT 1  
 FROM bookings b2  
 INNER JOIN rooms r2 ON b2.room\_id = r2.room\_id  
 WHERE b2.guest\_id = g.guest\_id  
 AND r2.category\_id IN (SELECT category\_id FROM room\_categories WHERE name = "Deluxe")  
);

This SQL query is selecting the first and last names of guests who have only booked either standard or deluxe rooms.

SELECT g.first\_name, g.last\_name, r.room\_number  
FROM guests g  
INNER JOIN bookings b ON g.guest\_id = b.guest\_id  
INNER JOIN rooms r ON b.room\_id = r.room\_id  
WHERE r.category\_id IN (SELECT category\_id FROM room\_categories WHERE name IN ("Standard", "Deluxe"))  
GROUP BY g.guest\_id  
HAVING *COUNT*(DISTINCT r.category\_id) = 1;

This SQL query is selecting the first and last names of guests who have booked standard rooms but not suites.

SELECT g.first\_name, g.last\_name  
FROM guests g  
WHERE *EXISTS* (  
 SELECT 1  
 FROM bookings b  
 INNER JOIN rooms r ON b.room\_id = r.room\_id  
 WHERE b.guest\_id = g.guest\_id  
 AND r.category\_id IN (SELECT category\_id FROM room\_categories WHERE name = "Standard")  
)  
AND NOT *EXISTS* (  
 SELECT 1  
 FROM bookings b  
 INNER JOIN rooms r ON b.room\_id = r.room\_id  
 WHERE b.guest\_id = g.guest\_id  
 AND r.category\_id IN (SELECT category\_id FROM room\_categories WHERE name = "Suite")  
);

This SQL query is selecting the first and last names and room numbers of guests who have booked standard rooms and a massage service.

SELECT g.first\_name, g.last\_name, r.room\_number  
FROM guests g  
INNER JOIN bookings b ON g.guest\_id = b.guest\_id  
INNER JOIN rooms r ON b.room\_id = r.room\_id  
INNER JOIN service\_bookings sb ON b.guest\_id = sb.guest\_id  
WHERE r.category\_id IN (SELECT category\_id FROM room\_categories WHERE name = "Standard")  
AND sb.service\_id IN (SELECT service\_id FROM services WHERE name = "Massage");

This SQL query is selecting the total price of all bills for a guest with an id of 123.

SELECT *SUM*(bi.amount) AS total\_price  
FROM bills bi  
INNER JOIN bookings b ON b.guest\_id = bi.guest\_id AND bi.date BETWEEN b.check\_in\_date AND b.check\_out\_date  
WHERE b.guest\_id = 123;  
  
ALTER TABLE reviews  
DROP COLUMN check\_in\_date;  
  
ALTER TABLE payments  
ADD COLUMN payment\_date;