

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO BÀI TẬP CUỐI KỲ

MÔN: THỰC HÀNH CƠ SỞ DỮ LIỆU

**CHỦ ĐỀ: XÂY DỰNG CƠ SỞ SỬ LIỆU CHO HỆ THỐNG
ĐẶT PHÒNG KHÁCH SẠN**

Mã học phần: IT3290

Mã lớp: 147779

Giảng viên hướng dẫn

Nhóm

Sinh viên thực hiện

Thầy Nguyễn Hữu Đức

4

Tạ Hồng Phúc – 20225906

Nguyễn Mạnh Tùng – 20225682

Bùi Quang Hưng – 20225849

Hà Nội, tháng 6 năm 2024



MỤC LỤC

I. Mục tiêu	3
1. Thiết kế cơ sở dữ liệu Website đặt phòng khách sạn	3
2. Xây dựng Website demo một số chức năng	3
II. Quy trình thiết kế	3
1. Phân tích các yêu cầu cần có của một Website đặt phòng	3
2. Khảo sát thực tế	3
3. Xác định các thực thể và thuộc tính	3
4. Sơ đồ thực thể liên kết	4
5. Sơ đồ quan hệ logic	Error! Bookmark not defined.
6. Chi tiết sơ đồ quan hệ	5
III. Câu lệnh SQL	6
1. Bảng nội dung yêu cầu các câu lệnh SQL	6
1.1. Truy vấn	6
1.2. View	6
1.3. Function	6
1.4. Trigger	7
1.5. Index	7
2. Câu lệnh SQL và kết quả	7
2.1. Truy vấn	7
2.2. View	13
2.3. Function	17
2.4. Trigger	25
2.5. Index	27
IV. Kết luận	28

I. Mục tiêu

1. Thiết kế cơ sở dữ liệu Website đặt phòng khách sạn
2. Xây dựng Website demo một số chức năng

II. Quy trình thiết kế

1. Phân tích các yêu cầu cần có của một Website đặt phòng

- Tra cứu phòng theo nhu cầu của khách hàng: tìm kiếm dựa trên các tiêu chí như loại phòng, giá phòng, rating,...
- Ghi nhận thông tin thuê phòng: ghi nhận thông tin chi tiết của khách hàng và yêu cầu thuê phòng.
- Ghi nhận thông tin check in, check out: lưu trữ thông tin về ngày giờ nhận và trả phòng.
- Cập nhật giá phòng, dịch vụ: Cho phép quản lý cập nhật giá phòng và các dịch vụ liên quan.
- Tính tiền phải thanh toán khi đặt phòng: Tính toán tổng số tiền phải thanh toán bao gồm giá và dịch vụ phụ theo.
- Cung cấp số liệu báo cáo: Cung cấp các báo cáo về tình trạng phòng, doanh thu,....
- Thêm, sửa, xóa các thông tin: Cho phép quản lý thực hiện thêm, sửa, xóa thông tin về khách sạn, phòng. Khách hàng có thể hủy đặt phòng.

2. Khảo sát thực tế

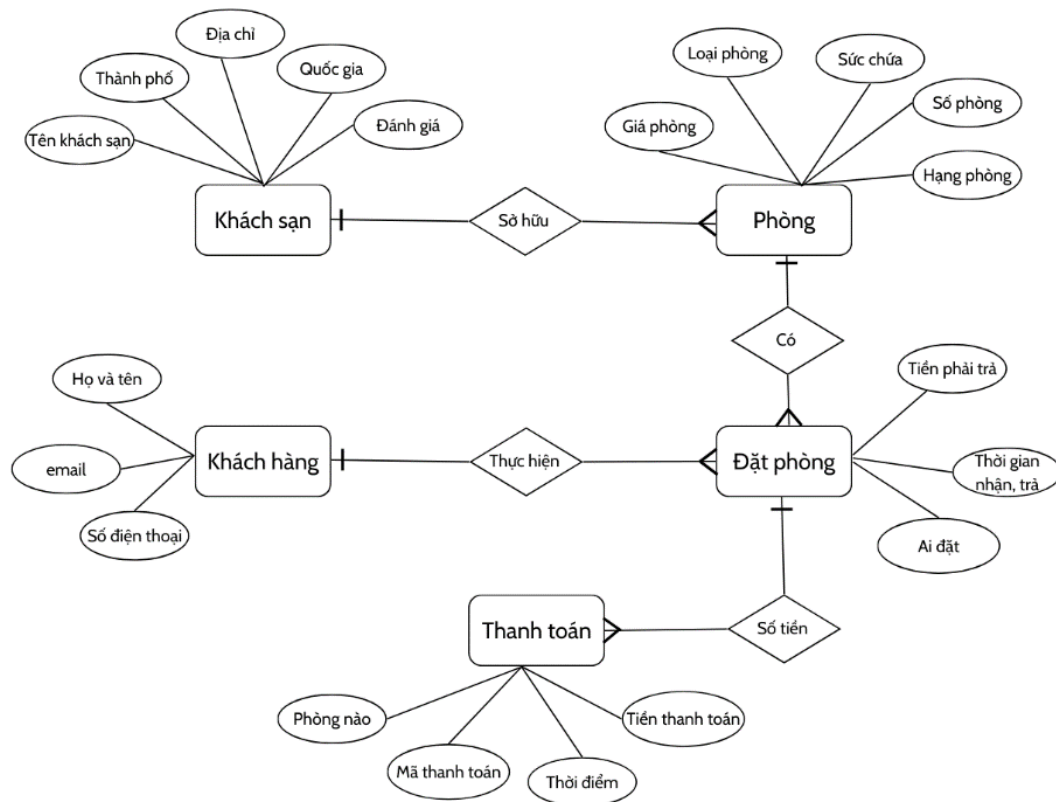
- Một khách sạn có nhiều loại phòng: đơn, đôi, gia đình, VIP,...
- Có nhiều khách sạn với các mức đánh giá khác nhau: 1, 2, 3, 4, 5 sao.
- Phòng thuê có thể có nhiều trạng thái: sẵn sàng, có khách, đã đặt trước, bảo trì,...
- Mỗi khách sạn có quy định, dịch vụ đi kèm khác nhau
- Thông tin khách hàng: cần ghi nhận lại thông tin liên lạc của khách hàng như họ tên, email, số điện thoại,...

3. Xác định các thực thể và thuộc tính

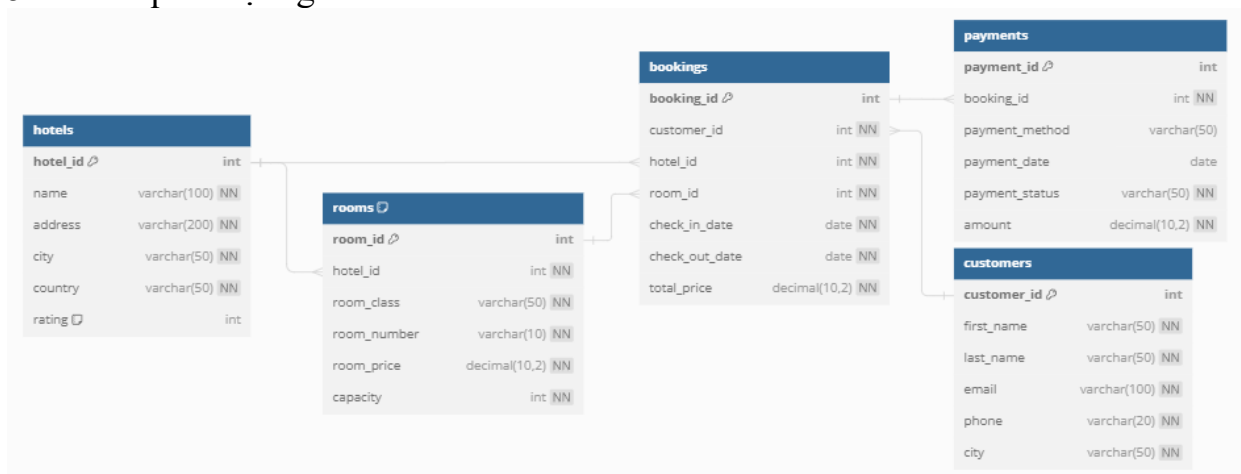
- Khách sạn: Tên, địa chỉ, đánh giá
- Phòng: Giá phòng, loại phòng, sức chứa, hạng phòng
- Khách hàng: Họ và tên, email, số điện thoại
- Đặt phòng: Thời gian check in – check out, tiền phải trả

- Thanh toán: Số tiền, thời điểm thanh toán

4. Sơ đồ thực thể liên kết



5. Sơ đồ quan hệ logic



6. Chi tiết sơ đồ quan hệ

a) hotels: Bảng khách sạn (lưu thông tin của khách sạn)

- hotel_id: khóa chính, đại diện cho mỗi khách sạn
- name: tên khách sạn
- address: địa chỉ khách sạn
- city: thành phố tọa lạc
- country: quốc gia tọa lạc
- rating: đánh giá của khách hàng (từ 1 đến 5 sao)

b) rooms: bảng các phòng trong khách sạn

- room_id: khóa chính, đại diện cho mỗi phòng
- hotel_id: mã khách sạn, tham chiếu đến hotels.hotel_id để biết thuộc khách sạn nào
- room_class: hạng phòng (Classic, Standard, Luxury)
- room_number: số phòng
- room_price: giá phòng
- capacity: sức chứa của phòng
- UNIQUE(hotel_id, room_number): Đảm bảo không có hai phòng cùng số trong một khách sạn

c) customers: bảng khách hàng

- customer_id: khóa chính, đại diện cho mỗi khách hàng
- first_name: họ của khách
- last_name: tên của khách hàng
- email: địa chỉ email
- phone: số điện thoại
- city: địa chỉ của khách hàng

d) bookings: bảng lưu thông tin đặt phòng

- booking_id: khóa chính, đại diện cho mỗi đơn đặt phòng
- customer_id: mã khách hàng, tham chiếu đến customers.customer_id để biết ai là người đặt phòng
- hotel_id: mã khách sạn, tham chiếu đến hotels.hotel_id để biết khách sạn được đặt
- room_id: mã phòng, tham chiếu đến rooms.room_id để biết phòng được đặt
- check_in_date: ngày nhận phòng
- check_out_date: ngày trả phòng

- `total_price`: tổng giá tiền cho đơn đặt phòng
- e) `payments`: bảng chứa thông tin thanh toán
- `payment_id`: khóa chính, đại diện cho mỗi đơn thanh toán
 - `booking_id` (INT NOT NULL REFERENCES `bookings(booking_id)` ON DELETE CASCADE): mã đặt phòng, tham chiếu đến `bookings.booking_id` để biết thanh toán cho phòng nào. Khi xóa đơn đặt phòng thì các thông tin thanh toán liên quan cũng bị xóa
 - `payment_method`: phương thức thanh toán (tiền mặt, thẻ tín dụng)
 - `payment_date`: ngày thanh toán
 - `payment_status`: trạng thái thanh toán của khách hàng
 - `amount`: số tiền thanh toán

III. Câu lệnh SQL

1. Bảng nội dung yêu cầu các câu lệnh SQL

1.1. Truy vấn

STT	Yêu cầu
1	Đưa ra tên khách sạn, địa chỉ, có phòng ‘Luxury’ rẻ nhất của từng thành phố
2	Đưa ra tên khách hàng ở Hà Nội, phòng khách sạn mà họ đã thuê
3	Liệt kê số phòng có nhiều hơn một lượt của từng khách sạn ở Hà Nội
4	Liệt kê phòng có ngày thuê dài nhất của mỗi khách sạn ở Hà Nội
5	Đưa ra tên khách sạn có lượng đặt phòng nhiều nhất
6	Thống kê số lượng khách sạn của mỗi tỉnh
7	Đưa ra các khách sạn có tổng doanh thu cao nhất trong năm hiện tại
8	Đưa ra các khách sạn có ít nhất một phòng ‘Luxury’ trống và đưa ra phòng đó
9	Thống kê tỉ lệ sử dụng theo từng phương thức thanh toán của khách hàng
10	Thống kê lượng đặt phòng (>2) theo từng tháng của mỗi khách sạn

1.2. View

STT	Yêu cầu
1	Danh sách các phòng đang được đặt
2	Các phòng được thanh toán gần đây (<10)
3	Tổng doanh thu cho từng khách sạn

1.3. Function

STT	Yêu cầu
1	Viết hàm khi nhập tên của khách hàng bất kì sẽ trả về lịch trình thuê phòng của khách hàng đó
2	Viết hàm tính tiền, khi nhập tên của khách hàng sẽ đưa ra tổng số tiền người đó đã chi



3	Viết hàm khi nhập tên của khách sạn sẽ thống kê tỉ lệ khách thuê phòng ‘Classic’, ‘Standard’, ‘Luxury’ của khách sạn đó
4	Viết hàm khi nhập tên khách sạn, đưa ra thông tin về khách hàng đã thuê phòng của khách sạn đó
5	Viết hàm đưa ra khách hàng đi du lịch từ tỉnh A đến tỉnh B
6	Viết hàm thống kê doanh thu theo tháng bất kì của khách sạn nào đó
7	Viết hàm khi nhập vào giá tiền sẽ đưa ra các phòng có mức giá bé hơn hoặc bằng giá đó
8	Viết hàm đưa ra khách hàng sử dụng thanh toán bằng phương thức X nhiều nhất
9	Viết hàm truy xuất danh sách các khách sạn và thông tin về các loại phòng có sẵn trong mỗi khách sạn
10	Viết hàm trả về danh sách các phòng có sẵn dựa trên các tham số đầu vào

1.4. Trigger

STT	Yêu cầu
1	Cập nhật giá tiền thanh toán khi có thay đổi (ngày checkout, giá của phòng,...)
2	Ngăn chặn đặt phòng quá mức
3	Khi bảng bookings được thêm bản ghi, thêm tương ứng bản ghi ở bảng payments với yêu cầu: <code>payment_method = NULL</code> , <code>payment_date = NULL</code> , <code>payment_status = ‘In Progress’</code> và <code>payments.amount = bookings.total_price</code>
4	Khi update <code>payment_date</code> , <code>payment_status = ‘Completed’</code>

1.5. Index

STT	Yêu cầu
1	Tăng tốc độ truy vấn khi tìm kiếm các khách sạn theo thành phố
2	Cải thiện hiệu suất khi tìm kiếm các đơn đặt phòng của một khách hàng cụ thể
3	Tìm kiếm các đặt phòng trong một khách sạn theo khoảng thời gian nhất định

2. Câu lệnh SQL và kết quả

2.1. Truy vấn

Câu 1: Đưa ra tên khách sạn, địa chỉ, có phòng ‘Luxury’ rẻ nhất của từng thành phố

SQL:



```

1 WITH cheapest_luxury_rooms AS (
2     SELECT
3         h.city,
4         MIN(r.room_price) AS min_price
5     FROM hotels h
6     INNER JOIN rooms r ON h.hotel_id = r.hotel_id
7     WHERE r.room_class = 'Luxury'
8     GROUP BY h.city
9 )
10
11 SELECT
12     h.name AS hotel_name,
13     h.address AS hotel_address,
14     clr.min_price AS cheapest_luxury_price
15 FROM hotels h
16 INNER JOIN rooms r ON h.hotel_id = r.hotel_id
17 INNER JOIN cheapest_luxury_rooms clr ON h.city = clr.city
18 WHERE r.room_class = 'Luxury' AND r.room_price = clr.min_price
19 ORDER BY h.city;

```

Kết quả:

	hotel_name character varying (100)	hotel_address character varying (200)	cheapest_luxury_price numeric
1	Khách sạn West Hotel	88-90-92 Hai Ba Trung	580.00
2	Khách sạn Monarque	236 Vo Nguyen Giap, Phuoc My, Son Tra	450.00
3	Khách sạn Pan Pacific Ha Noi	1 Thanh Nien, Truc Bach, Ba Dinh	500.00
4	Khách sạn Melia Ha Noi	44B Ly Thuong Kiet, Tran Hung Dao, Hoan Kiem	500.00

Câu 2: Đưa ra thông tin khách hàng ở Hà Nội, phòng khách sạn mà họ đã thuê

SQL:

```

1 SELECT
2     c.customer_id,
3     c.first_name || ' ' || c.last_name AS customer_name,
4     c.email,
5     c.phone,
6     h.name AS hotel_name,
7     r.room_number,
8     b.check_in_date,
9     b.check_out_date
10 FROM customers c
11 JOIN bookings b ON c.customer_id = b.customer_id
12 JOIN rooms r ON b.room_id = r.room_id
13 JOIN hotels h ON b.hotel_id = h.hotel_id
14 WHERE c.city = 'Ha Noi';

```

Kết quả:

	customer_id integer	customer_name text	email character varying (100)	phone character varying (20)	hotel_name character varying (100)	room_number character varying (10)	check_in_date date	check_out_date date
1	1	Nguyen Van A	nguyenvana@example.com	0901234567	Khách sạn Melia Ha Noi	101	2024-05-01	2024-05-03
2	1	Nguyen Van A	nguyenvana@example.com	0901234567	Khách sạn Melia Ha Noi	102	2024-06-10	2024-06-12
3	68	Ngo Thi PPP	ngothiPPP@example.com	0901234534	Khách sạn Queen Ann	101	2024-07-28	2024-08-02



Câu 3: Liệt kê số phòng có nhiều hơn một lượt của từng khách sạn ở Hà Nội

SQL:

```

1  SELECT
2      h.name AS hotel_name,
3      r.room_number,
4      COUNT(b.booking_id) AS booking_count
5  FROM hotels h
6  JOIN rooms r ON h.hotel_id = r.hotel_id
7  JOIN bookings b ON r.room_id = b.room_id
8  JOIN customers c ON b.customer_id = c.customer_id
9  WHERE h.city = 'Ha Noi'
10 GROUP BY h.name, r.room_number
11 HAVING COUNT(b.booking_id) > 1
12 ORDER BY h.name, booking_count DESC;

```

Kết quả:

	hotel_name character varying (100)	room_number character varying (10)	booking_count bigint
1	Khách sạn JW Marriott Ha Noi	101	3
2	Khách sạn Lotte Ha Noi	101	3
3	Khách sạn Lotte Ha Noi	102	2
4	Khách sạn Melia Ha Noi	101	3
5	Khách sạn Melia Ha Noi	102	2

Câu 4: Liệt kê phòng có ngày thuê dài nhất (>1) của mỗi khách sạn ở Hà Nội

SQL:

```

1  WITH LongestBookedRooms AS (
2      SELECT r.room_id, r.hotel_id,
3             (b.check_out_date - b.check_in_date) AS booking_duration,
4             ROW_NUMBER() OVER(PARTITION BY r.hotel_id ORDER BY (b.check_out_date - b.check_in_date) DESC) AS rn
5  FROM rooms r
6  JOIN bookings b ON r.room_id = b.room_id
7  JOIN hotels h ON r.hotel_id = h.hotel_id
8  WHERE h.city = 'Ha Noi'
9  ),
10 MaxDurationPerHotel AS (
11     SELECT hotel_id, room_id, booking_duration
12     FROM LongestBookedRooms
13     WHERE rn = 1
14 )
15 SELECT h.name AS hotel_name, h.address AS hotel_address, r.room_number, m.booking_duration AS max_booking_duration
16 FROM MaxDurationPerHotel m
17 JOIN rooms r ON m.room_id = r.room_id
18 JOIN hotels h ON m.hotel_id = h.hotel_id
19 ORDER BY hotel_name;

```

Kết quả:



	hotel_name character varying (100)	hotel_address character varying (200)	room_number character varying (10)	max_booking_duration integer
1	Khách sạn JW Marriott Ha Noi	8 Do Duc Duc, Me Tri, Nam Tu Liem	202	4
2	Khách sạn Lotte Ha Noi	54 Lieu Giai, Cong Vi, Ba Dinh	102	2
3	Khách sạn Melia Ha Noi	44B Ly Thuong Kiet, Tran Hung Dao, Hoan Kiem	201	3
4	Khách sạn Pan Pacific Ha Noi	1 Thanh Nien, Truc Bach, Ba Dinh	101	2
5	Khách sạn Sofitel Legend Metropole	15 Ngo Quyen, Hoan Kiem	201	2

Câu 5: Đưa ra tên khách sạn có lượng đặt phòng nhiều nhất

SQL:

```

1 SELECT
2     h.name AS hotel_name,
3     COUNT(b.booking_id) AS booking_count
4 FROM hotels h
5 JOIN rooms r ON h.hotel_id = r.hotel_id
6 JOIN bookings b ON r.room_id = b.room_id
7 GROUP BY h.name
8 ORDER BY booking_count DESC
9 LIMIT 1;

```

Kết quả:

	hotel_name character varying (100)	booking_count bigint
1	Khách sạn Melia Ha Noi	8

Câu 6: Thống kê số lượng khách sạn của mỗi tỉnh

SQL:

```

1 SELECT city, COUNT(*) AS hotel_count
2 FROM hotels
3 GROUP BY city
4 ORDER BY hotel_count DESC;

```

Kết quả:

	city character varying (50)	hotel_count bigint
1	Nha Trang	5
2	Da Nang	5
3	Can Tho	5
4	Hue	5
5	Ho Chi Minh	5
6	Hai Phong	5
7	Ha Noi	5

Câu 7: Đưa ra khách sạn có tổng doanh thu cao nhất trong năm hiện tại

SQL:

```
1 WITH HotelRevenues AS (  
2     SELECT  
3         h.hotel_id,  
4         h.name AS hotel_name,  
5         h.city AS hotel_city,  
6         SUM(p.amount) AS total_revenue,  
7         EXTRACT(YEAR FROM b.check_in_date) AS booking_year  
8     FROM  
9         hotels h  
10        JOIN bookings b ON h.hotel_id = b.hotel_id  
11        JOIN payments p ON b.booking_id = p.booking_id  
12    WHERE  
13        EXTRACT(YEAR FROM b.check_in_date) = EXTRACT(YEAR FROM CURRENT_DATE)  
14    GROUP BY  
15        h.hotel_id, h.name, booking_year  
16 )  
17 SELECT  
18     hotel_id,  
19     hotel_name,  
20     hotel_city,  
21     total_revenue  
22 FROM  
23     HotelRevenues  
24 WHERE  
25     total_revenue = (SELECT MAX(total_revenue) FROM HotelRevenues);
```

Kết quả:

	hotel_id [PK] integer	hotel_name character varying (100)	hotel_city character varying (50)	total_revenue numeric
1	9	Khách sạn Lavis 18 Residence	Hồ Chí Minh	5600.00

Câu 8: Đưa ra các khách sạn ở Hải Phòng có phòng ‘Luxury’ trống và đưa ra phòng đó

SQL:



```

1  SELECT
2      h.hotel_id,
3      h.name AS hotel_name,
4      r.room_id,
5      r.room_number,
6      r.room_price,
7      r.room_class
8  FROM
9      hotels h
10 JOIN
11     rooms r ON h.hotel_id = r.hotel_id
12 LEFT JOIN
13     bookings b ON r.room_id = b.room_id
14                AND CURRENT_DATE BETWEEN b.check_in_date AND b.check_out_date
15 WHERE
16     h.city = 'Hai Phong'
17     AND r.room_class = 'Luxury'
18     AND b.booking_id IS NULL;

```

Kết quả:

	hotel_id integer	hotel_name character varying (100)	room_id integer	room_number character varying (10)	room_price numeric (10,2)	room_class character varying (50)
1	33	Khách sạn Pearl River Hai Phong	165	301	600.00	Luxury
2	35	Khách sạn Avani Hai Phong Harbour View	175	301	500.00	Luxury
3	31	Khách sạn Somerset Central TD Hai Phong	155	301	650.00	Luxury

Câu 9: Thống kê tỉ lệ sử dụng theo từng phương thức thanh toán của khách hàng

SQL:

```

1  SELECT
2      payment_method,
3      COUNT(*) AS payment_count,
4      ROUND(COUNT(*) * 100.0 / SUM(COUNT(*)) OVER(), 2) AS percentage
5  FROM payments
6  GROUP BY payment_method
7  ORDER BY payment_count DESC;

```

Kết quả:

	payment_method character varying (50)	payment_count bigint	percentage numeric
1	Internet Banking	23	27.71
2	Card	21	25.30
3	QR Pay	20	24.10
4	Momo	19	22.89

Câu 10: Thống kê lượng đặt phòng (>2) theo từng tháng của mỗi khách sạn

SQL:

```
1 SELECT
2     h.name AS hotel_name,
3     DATE_PART('month', b.check_in_date) AS month,
4     COUNT(*) AS bookings_count
5 FROM
6     hotels h
7 JOIN
8     bookings b ON h.hotel_id = b.hotel_id
9 GROUP BY
10    h.name, DATE_PART('month', b.check_in_date)
11 HAVING
12     COUNT(*) > 2
13 ORDER BY
14     hotel_name, month;
```

Kết quả:

	hotel_name character varying (100)	month double precision	bookings_count bigint
1	Khách sạn JW Marriott Ha Noi	6	4
2	Khách sạn Lavis 18 Residence	7	4
3	Khách sạn Lotte Ha Noi	6	4
4	Khách sạn Melia Ha Noi	6	5
5	Khách sạn Pan Pacific Ha Noi	6	4
6	Khách sạn Pullman Sai Gon Centre	7	4
7	Khách sạn Queen Ann	7	4
8	Khách sạn Rex	7	4
9	Khách sạn Silerland & Spa	8	4
10	Khách sạn Sofitel Legend Metropole	6	4

2.2. View

Câu 1: Danh sách các phòng đã được đặt trong tháng 7 của các khách sạn tại Hà Nội

SQL:



```
1 CREATE OR REPLACE VIEW booked_rooms AS
2 SELECT
3     r.room_id,
4     r.room_number,
5     r.room_class,
6     r.room_price,
7     h.hotel_id,
8     h.name AS hotel_name,
9     b.booking_id,
10    b.customer_id,
11    c.first_name AS customer_first_name,
12    c.last_name AS customer_last_name,
13    b.check_in_date,
14    b.check_out_date
15 FROM
16     rooms r
17 JOIN
18     bookings b ON r.room_id = b.room_id
19 JOIN
20     hotels h ON r.hotel_id = h.hotel_id
21 JOIN
22     customers c ON b.customer_id = c.customer_id
23 WHERE
24     h.city = 'Ha Noi'
25     AND EXTRACT(MONTH FROM b.check_in_date) = 7;
```

Kết quả: SELECT * FROM booked_rooms;

		room_number character varying (10)	room_class character varying (50)	room_price numeric (10,2)	hotel_id integer	hotel_name character varying (100)	booking_id integer	customer_id integer	customer_first_name character varying (50)	customer_last_name character varying (50)	check_in_date date	check_out_date date
1	5	301	Luxury	500.00	1	Khách sạn Melia Ha Noi	157	71	Pham	Van SSS	2024-07-11	2024-07-14
2	7	102	Classic	120.00	2	Khách sạn Lotte Ha Noi	87	2	Tran	Thi B	2024-07-01	2024-07-03
3	10	301	Standard	300.00	2	Khách sạn Lotte Ha Noi	158	72	Bui	Thi TTT	2024-07-12	2024-07-13
4	15	301	Luxury	600.00	3	Khách sạn Sofitel Legend Metropole	159	73	Nguyen	Van UUU	2024-07-13	2024-07-15
5	20	301	Standard	250.00	4	Khách sạn JW Marriott Ha Noi	160	74	Le	Thi VVV	2024-07-14	2024-07-18
6	25	301	Luxury	500.00	5	Khách sạn Pan Pacific Ha Noi	161	75	Tran	Van WWW	2024-07-15	2024-07-16

Câu 2: Các phòng được thanh toán trong vòng một tuần gần đây ở Nha Trang



SQL:

```
1 CREATE OR REPLACE VIEW paid_rooms_recent AS
2 SELECT
3     r.room_id,
4     r.room_number,
5     r.room_price,
6     h.hotel_id,
7     h.name AS hotel_name,
8     h.address AS hotel_address,
9     h.city AS hotel_city,
10    h.country AS hotel_country,
11    p.payment_id,
12    p.payment_method,
13    p.payment_date,
14    p.payment_status,
15    p.amount
16 FROM
17     rooms r
18 JOIN
19     bookings b ON r.room_id = b.room_id
20 JOIN
21     hotels h ON r.hotel_id = h.hotel_id
22 JOIN
23     payments p ON b.booking_id = p.booking_id
24 WHERE
25     p.payment_date >= CURRENT_DATE - INTERVAL '7 days' -- Thanh toán trong 7 ngày gần đây
26     AND r.capacity < 10
27     AND h.city = 'Nha Trang';
```

Kết quả: SELECT * FROM paid_rooms_recent;

payment_id integer	payment_method character varying (50)	payment_date date	payment_status character varying (50)	amount numeric (10,2)
18	QR Pay	2024-06-19	Completed	460.00
19	Card	2024-06-18	Completed	400.00
20	Momo	2024-06-21	Completed	570.00
21	Internet Banking	2024-06-20	Completed	800.00
22	QR Pay	2024-07-23	Completed	360.00
23	Card	2024-07-27	Completed	360.00

Câu 3: Tổng doanh thu của 5 khách sạn cao nhất

SQL:



```
1 CREATE OR REPLACE VIEW top_5_hotel_revenue AS
2 WITH hotel_revenue_summary AS (
3     SELECT
4         h.hotel_id,
5         h.name AS hotel_name,
6         SUM(p.amount) AS total_revenue,
7         RANK() OVER (ORDER BY SUM(p.amount) DESC) AS revenue_rank
8     FROM
9         hotels h
10    JOIN
11        bookings b ON h.hotel_id = b.hotel_id
12    JOIN
13        payments p ON b.booking_id = p.booking_id
14    GROUP BY
15        h.hotel_id, h.name
16 )
17 SELECT
18     hotel_id,
19     hotel_name,
20     total_revenue,
21     revenue_rank
22 FROM
23     hotel_revenue_summary
24 WHERE
25     revenue_rank <= 5
26 ORDER BY
27     total_revenue DESC;
```

Kết quả: SELECT * FROM top_5_hotel_revenue;

	hotel_id integer	hotel_name character varying (100)	total_revenue numeric	revenue_rank bigint
1	9	Khách sạn Lavis 18 Residence	5600.00	1
2	3	Khách sạn Sofitel Legend Metropole	4900.00	2
3	8	Khách sạn Queen Ann	4685.00	3
4	4	Khách sạn JW Marriott Ha Noi	4375.00	4
5	1	Khách sạn Melia Ha Noi	4000.00	5



2.3. Function

Câu 1: Viết hàm khi nhập tên của khách hàng bất kì sẽ trả về lịch trình thuê phòng của khách hàng đó

SQL:

```

1 CREATE OR REPLACE FUNCTION get_customer_booking_schedule_by_last_name(p_last_name VARCHAR(50))
2 RETURNS TABLE (
3     booking_id INT,
4     hotel_name VARCHAR,
5     room_class VARCHAR,
6     check_in_date DATE,
7     check_out_date DATE,
8     total_price DECIMAL(10, 2)
9 )
10 AS $$
11 BEGIN
12     RETURN QUERY
13     SELECT
14         b.booking_id,
15         h.name AS hotel_name,
16         r.room_class,
17         b.check_in_date,
18         b.check_out_date,
19         b.total_price
20     FROM
21         customers c
22     JOIN
23         bookings b ON c.customer_id = b.customer_id
24     JOIN
25         hotels h ON b.hotel_id = h.hotel_id
26     JOIN
27         rooms r ON b.room_id = r.room_id
28     WHERE
29         c.last_name = p_last_name;
30 END;
31 $$
32 LANGUAGE plpgsql;

```

Kết quả: SELECT * FROM get_customer_booking_schedule_by_last_name('Van A');

	booking_id integer	hotel_name character varying	room_class character varying	check_in_date date	check_out_date date	total_price numeric
1	84	Khách sạn Melia Hà Nội	Standard	2024-05-01	2024-05-03	300.00
2	85	Khách sạn Melia Hà Nội	Standard	2024-06-10	2024-06-12	300.00

Câu 2: Viết hàm tính tiền, khi nhập tên của khách hàng sẽ đưa ra tổng số tiền người đó đã chi

SQL:



```

1 CREATE OR REPLACE FUNCTION calculate_total_spent_by_customer(
2   customer_first_name character varying,
3   customer_last_name character varying
4 )
5 RETURNS TABLE (
6   hotel_name VARCHAR(100),
7   total_spent DECIMAL(10, 2)
8 )
9 AS $$
10 BEGIN
11   RETURN QUERY
12   SELECT h.name AS hotel_name, SUM(b.total_price) AS total_spent
13   FROM bookings b
14   JOIN customers c ON b.customer_id = c.customer_id
15   JOIN hotels h ON b.hotel_id = h.hotel_id
16   WHERE c.first_name = customer_first_name
17         AND c.last_name = customer_last_name
18   GROUP BY h.name;
19 END;
20 $$ LANGUAGE plpgsql;

```

Kết quả: `SELECT * FROM calculate_total_spent_by_customer('Nguyen', 'Van A');`

	hotel_name character varying	total_spent numeric
1	Khách sạn Melia Ha Noi	600.00

Câu 3: Viết hàm khi nhập tên của khách sạn sẽ thống kê tỉ lệ khách thuê phòng ‘Classic’, ‘Standard’, ‘Luxury’ của khách sạn đó

SQL:

```

1 CREATE OR REPLACE FUNCTION calculate_room_type_ratio(
2   hotel_name VARCHAR
3 )
4 RETURNS TABLE (
5   room_class VARCHAR(50),
6   booking_count BIGINT,
7   total_bookings BIGINT,
8   booking_ratio DECIMAL(5, 2)
9 )
10 AS $$
11 BEGIN
12   RETURN QUERY
13   SELECT
14     r.room_class,
15     COUNT(b.booking_id) AS booking_count,
16     SUM(COUNT(b.booking_id) OVER ()):BIGINT AS total_bookings,
17     ROUND(COUNT(b.booking_id) * 100.0 / SUM(COUNT(b.booking_id) OVER ()), 2) AS booking_ratio
18   FROM bookings b
19   JOIN hotels h ON b.hotel_id = h.hotel_id
20   JOIN rooms r ON b.room_id = r.room_id
21   WHERE h.name = hotel_name
22         AND r.room_class IN ('Classic', 'Standard', 'Luxury')
23   GROUP BY r.room_class
24   ORDER BY r.room_class;
25 END;
26 $$ LANGUAGE plpgsql;

```



Kết quả: `SELECT * FROM calculate_room_type_ratio('Khach san Rex');`

	room_class character varying	booking_count bigint	total_bookings bigint	booking_ratio numeric
1	Classic	4	7	57.14
2	Standard	3	7	42.86

Câu 4: Viết hàm khi nhập tên khách sạn, đưa ra thông tin về khách hàng đã thuê phòng của khách sạn đó

SQL:

```

1 CREATE OR REPLACE FUNCTION get_customer_booking_info(
2     hotel_name VARCHAR(100)
3 )
4 RETURNS TABLE (
5     customer_id INT,
6     first_name VARCHAR(100),
7     last_name VARCHAR(100),
8     check_in_date DATE,
9     check_out_date DATE
10 )
11 AS $$
12 BEGIN
13     RETURN QUERY
14     SELECT
15         c.customer_id,
16         c.first_name,
17         c.last_name,
18         b.check_in_date,
19         b.check_out_date
20     FROM customers c
21     JOIN bookings b ON c.customer_id = b.customer_id
22     JOIN hotels h ON b.hotel_id = h.hotel_id
23     WHERE h.name = hotel_name;
24 END;
25 $$ LANGUAGE plpgsql;

```

Kết quả: `SELECT * FROM get_customer_booking_info('Khach san Rex');`

	customer_id integer	first_name character varying	last_name character varying	check_in_date date	check_out_date date
1	6	Vu	Thi F	2024-06-06	2024-06-09
2	26	Vu	Thi Z	2024-06-26	2024-06-29
3	36	Dinh	Thi JJ	2024-07-06	2024-07-10
4	46	Vu	Thi TT	2024-07-06	2024-07-09
5	56	Dinh	Thi DDD	2024-07-16	2024-07-20
6	66	Vu	Thi NNN	2024-07-26	2024-07-29
7	76	Dinh	Thi XXX	2024-08-16	2024-08-20



Câu 5: Viết hàm đưa ra khách hàng đi du lịch từ tỉnh A đến tỉnh B

SQL:

```

1 CREATE OR REPLACE FUNCTION get_customers_travel_from_to(province_a VARCHAR, province_b VARCHAR)
2 RETURNS TABLE (
3     first_name VARCHAR,
4     last_name VARCHAR,
5     home_address VARCHAR,
6     travel_address VARCHAR
7 ) AS $$
8 BEGIN
9     RETURN QUERY
10    SELECT
11        c.first_name,
12        c.last_name,
13        c.city AS home_address,
14        h.city AS travel_address
15    FROM customers c
16    JOIN bookings b ON c.customer_id = b.customer_id
17    JOIN hotels h ON b.hotel_id = h.hotel_id
18    WHERE c.city = province_a
19    AND h.city = province_b
20    ORDER BY c.first_name, c.last_name;
21 END;
22 $$ LANGUAGE plpgsql;

```

Kết quả: SELECT * FROM get_customers_travel_from_to('Ha Noi', 'Ho Chi Minh');

	first_name character varying	last_name character varying	home_address character varying	travel_address character varying
1	Ngo	Thi PPP	Ha Noi	Ho Chi Minh

Câu 6: Viết hàm thống kê doanh thu theo tháng bất kì của khách sạn nào đó

SQL:

```

1 CREATE OR REPLACE FUNCTION calculate_monthly_revenue_by_name_and_month(
2     hotel_name VARCHAR(100),
3     target_month INT,
4     target_year INT
5 )
6 RETURNS TABLE(revenue DECIMAL(10, 2)) AS $$
7 BEGIN
8     RETURN QUERY
9     SELECT
10        SUM(p.amount) AS revenue
11    FROM
12        bookings b
13        JOIN payments p ON b.booking_id = p.booking_id
14        JOIN hotels h ON b.hotel_id = h.hotel_id
15    WHERE
16        h.name = hotel_name
17        AND EXTRACT(MONTH FROM b.check_in_date) = target_month
18        AND EXTRACT(YEAR FROM b.check_in_date) = target_year;
19 END;
20 $$ LANGUAGE plpgsql;

```



Kết quả: `SELECT * FROM`

`calculate_monthly_revenue_by_name_and_month('Khach san Rex', 6, 2024);`

	revenue
1	660.00

Câu 7: Viết hàm khi nhập vào giá tiền sẽ đưa ra các phòng có mức giá bé hơn hoặc bằng giá đó

SQL:

```

1 CREATE OR REPLACE FUNCTION get_rooms_by_price(max_price DECIMAL(10, 2))
2 RETURNS TABLE (
3     room_id INT,
4     room_number VARCHAR(10),
5     room_price DECIMAL(10, 2),
6     hotel_id INT
7 ) AS $$
8 BEGIN
9     RETURN QUERY
10    SELECT
11        r.room_id,
12        r.room_number,
13        r.room_price,
14        r.hotel_id
15    FROM
16        rooms r
17    WHERE
18        r.room_price <= max_price;
19 END;
20 $$ LANGUAGE plpgsql;

```

Kết quả: `SELECT * FROM get_rooms_by_price(90);`

	room_id	room_number	room_price	hotel_id
1	26	101	90.00	6
2	27	102	90.00	6
3	46	101	80.00	10
4	47	102	80.00	10
5	126	101	90.00	26
6	127	102	90.00	26



Câu 8: Viết hàm đưa ra tỉ lệ khách hàng sử dụng phương thức thanh toán X

SQL:

```

1 CREATE OR REPLACE FUNCTION calculate_payment_method_ratio(
2   payment_method_name VARCHAR
3 )
4 RETURNS TABLE (
5   payment_method VARCHAR(50),
6   customer_count BIGINT,
7   total_customers BIGINT,
8   customer_ratio DECIMAL(5, 2)
9 )
10 AS $$
11 BEGIN
12   RETURN QUERY
13   SELECT
14     p.payment_method,
15     COUNT(DISTINCT b.customer_id) AS customer_count,
16     (SELECT COUNT(DISTINCT customer_id) FROM bookings)::BIGINT AS total_customers,
17     ROUND(COUNT(DISTINCT b.customer_id) * 100.0 / (SELECT COUNT(DISTINCT customer_id) FROM bookings), 2) AS customer_ratio
18   FROM payments p
19   JOIN bookings b ON p.booking_id = b.booking_id
20   WHERE p.payment_method = payment_method_name
21   GROUP BY p.payment_method;
22 END;
23 $$ LANGUAGE plpgsql;

```

Kết quả; SELECT * FROM calculate_payment_method_ratio('Internet Banking');

	payment_method character varying	customer_count bigint	total_customers bigint	customer_ratio numeric
1	Internet Banking	22	80	27.50

Câu 9: Viết hàm truy xuất danh sách các khách sạn và thông tin về các loại phòng có sẵn trong mỗi khách sạn

SQL:

```

1 CREATE OR REPLACE FUNCTION get_distinct_hotels_and_rooms()
2 RETURNS TABLE (
3   hotel_id INT,
4   hotel_name VARCHAR(100),
5   hotel_address VARCHAR(200),
6   hotel_city VARCHAR(50),
7   hotel_country VARCHAR(50),
8   hotel_rating INT,
9   room_class VARCHAR(50),
10  room_price DECIMAL(10, 2),
11  room_capacity INT
12 )
13 AS $$
14 BEGIN
15   RETURN QUERY
16   SELECT DISTINCT ON (h.hotel_id, r.room_class)

```

```
17         h.hotel_id,  
18         h.name AS hotel_name,  
19         h.address AS hotel_address,  
20         h.city AS hotel_city,  
21         h.country AS hotel_country,  
22         h.rating AS hotel_rating,  
23         r.room_class,  
24         r.room_price,  
25         r.capacity  
26     FROM  
27         hotels h  
28         JOIN rooms r ON h.hotel_id = r.hotel_id  
29     ORDER BY  
30         h.hotel_id, r.room_class;  
31 END;  
32 $$ LANGUAGE plpgsql;
```

Kết quả: SELECT * FROM get_distinct_hotels_and_rooms();

	hotel_id integer	hotel_name character varying	hotel_address character varying	hotel_city character varying	hotel_country character varying	hotel_rating integer	room_class character varying
1	1	Khách sạn Melia Ha Noi	44B Ly Thuong Kiet, Tran Hung Dao, Hoan Kiem	Ha Noi	Vietnam	4	Luxury
2	1	Khách sạn Melia Ha Noi	44B Ly Thuong Kiet, Tran Hung Dao, Hoan Kiem	Ha Noi	Vietnam	4	Standard
3	2	Khách sạn Lotte Ha Noi	54 Lieu Giai, Cong Vi, Ba Dinh	Ha Noi	Vietnam	4	Classic
4	2	Khách sạn Lotte Ha Noi	54 Lieu Giai, Cong Vi, Ba Dinh	Ha Noi	Vietnam	4	Standard
5	3	Khách sạn Sofitel Legend Metropole	15 Ngo Quyen, Hoan Kiem	Ha Noi	Vietnam	4	Luxury
6	3	Khách sạn Sofitel Legend Metropole	15 Ngo Quyen, Hoan Kiem	Ha Noi	Vietnam	4	Standard
7	4	Khách sạn JW Marriott Ha Noi	8 Do Duc Duc, Me Tri, Nam Tu Liem	Ha Noi	Vietnam	5	Classic
8	4	Khách sạn JW Marriott Ha Noi	8 Do Duc Duc, Me Tri, Nam Tu Liem	Ha Noi	Vietnam	5	Standard
9	5	Khách sạn Pan Pacific Ha Noi	1 Thanh Nien, Truc Bach, Ba Dinh	Ha Noi	Vietnam	5	Luxury
10	5	Khách sạn Pan Pacific Ha Noi	1 Thanh Nien, Truc Bach, Ba Dinh	Ha Noi	Vietnam	5	Standard
11	6	Khách sạn Rex	141 Nguyen Hue, Ben Nghe, Quan 1	Ho Chi Minh	Vietnam	3	Classic
12	6	Khách sạn Rex	141 Nguyen Hue, Ben Nghe, Quan 1	Ho Chi Minh	Vietnam	3	Standard
13	7	Khách sạn Pullman Sai Gon Centre	148 Tran Hung Dao, Quan 1	Ho Chi Minh	Vietnam	4	Luxury
14	7	Khách sạn Pullman Sai Gon Centre	148 Tran Hung Dao, Quan 1	Ho Chi Minh	Vietnam	4	Standard
15	8	Khách sạn Queen Ann	90 Nguyen Thi Minh Khai, Quan 3	Ho Chi Minh	Vietnam	4	Classic
16	8	Khách sạn Queen Ann	90 Nguyen Thi Minh Khai, Quan 3	Ho Chi Minh	Vietnam	4	Standard
17	9	Khách sạn Lavis 18 Residence	3 Do Thanh, Quan 3	Ho Chi Minh	Vietnam	5	Luxury
18	9	Khách sạn Lavis 18 Residence	3 Do Thanh, Quan 3	Ho Chi Minh	Vietnam	5	Standard
19	10	Khách sạn Silerland & Spa	10A Le Thanh Ton, Ben Nghe, Quan 1	Ho Chi Minh	Vietnam	3	Classic
20	10	Khách sạn Silerland & Spa	10A Le Thanh Ton, Ben Nghe, Quan 1	Ho Chi Minh	Vietnam	3	Standard
21	11	Khách sạn InterContinental Da Nang	Bai Bac, Son Tra	Da Nang	Vietnam	5	Luxury
22	11	Khách sạn InterContinental Da Nang	Bai Bac, Son Tra	Da Nang	Vietnam	5	Standard
23	12	Khách sạn Sala Danang Beach	36-38 Lam Hoanh, Phuoc My, Son Tra	Da Nang	Vietnam	4	Classic
24	12	Khách sạn Sala Danang Beach	36-38 Lam Hoanh, Phuoc My, Son Tra	Da Nang	Vietnam	4	Standard
25	13	Khách sạn Cicilia & Spa	Bai Bien T20, Phuong My An, Ngu Hanh Son	Da Nang	Vietnam	4	Luxury
26	13	Khách sạn Cicilia & Spa	Bai Bien T20, Phuong My An, Ngu Hanh Son	Da Nang	Vietnam	4	Standard
27	14	Khách sạn Maximilian Danang	222 Vo Nguyen Giap, Phuoc My, Son Tra	Da Nang	Vietnam	3	Classic
28	14	Khách sạn Maximilian Danang	222 Vo Nguyen Giap, Phuoc My, Son Tra	Da Nang	Vietnam	3	Standard



Câu 10: Viết hàm trả về danh sách các phòng có sẵn dựa trên các tham số đầu vào SQL:

```

1 CREATE OR REPLACE FUNCTION get_available_room(
2     p_hotel_id INT,
3     p_room_class VARCHAR(50),
4     p_check_in_date DATE,
5     p_check_out_date DATE
6 )
7 RETURNS TABLE (
8     room_id INT,
9     room_number VARCHAR(10),
10    room_class VARCHAR(50),
11    room_price DECIMAL(10, 2),
12    hotel_id INT,
13    capacity INT
14 )
15 AS $$
16 BEGIN
17     RETURN QUERY
18     SELECT
19         r.room_id,
20         r.room_number,
21         r.room_class,
22         r.room_price,
23         r.hotel_id,
24         r.capacity
25     FROM
26         rooms r
27     WHERE
28         r.hotel_id = p_hotel_id
29         AND r.room_class = p_room_class
30         AND r.room_id NOT IN (
31             SELECT
32                 b.room_id
33             FROM
34                 bookings b
35             WHERE
36                 (b.check_in_date >= p_check_in_date AND b.check_in_date < p_check_out_date)
37                 OR (b.check_out_date > p_check_in_date AND b.check_out_date <= p_check_out_date)
38                 OR (b.check_in_date < p_check_in_date AND b.check_out_date > p_check_out_date)
39         );
40 END;
41 $$ LANGUAGE plpgsql;

```

Kết quả: `SELECT * FROM get_available_room(1, 'Luxury', '2025-08-20', '2025-08-25');`

	room_id integer	room_number character varying	room_class character varying	room_price numeric	hotel_id integer	capacity integer
1	5	301	Luxury	500.00	1	6



2.4. Trigger

Câu 1: Đảm bảo khi có thay đổi về ngày check in, check out thì giá tiền vẫn luôn đúng

SQL:

```
1 CREATE OR REPLACE FUNCTION trg_update_prices()
2 RETURNS TRIGGER AS $$
3 BEGIN
4     UPDATE bookings
5     SET total_price = (NEW.check_out_date - NEW.check_in_date) * rooms.room_price
6     FROM rooms
7     WHERE bookings.room_id = rooms.room_id
8     AND bookings.booking_id = NEW.booking_id;
9     UPDATE payments
10    SET amount = bookings.total_price
11    FROM bookings
12    WHERE payments.booking_id = bookings.booking_id
13    AND payments.booking_id = NEW.booking_id;
14    RETURN NEW;
15 END;
16 $$ LANGUAGE plpgsql;
17 CREATE TRIGGER update_prices
18 AFTER UPDATE OF check_in_date, check_out_date ON bookings
19 FOR EACH ROW
20 WHEN (OLD.check_in_date IS DISTINCT FROM NEW.check_in_date OR OLD.check_out_date IS DISTINCT FROM NEW.check_out_date)
21 EXECUTE FUNCTION trg_update_prices();
```

Kết quả:

```
1 --UPDATE bookings SET check_out_date = '2024-08-09' WHERE booking_id = 167;
2 SELECT b.booking_id, b.customer_id, b.room_id, b.check_in_date, b.check_out_date,
3        p.payment_id, p.payment_method, p.payment_date, p.payment_status, p.amount
4 FROM bookings b
5 JOIN payments p ON b.booking_id = p.booking_id
6 WHERE b.booking_id = 167;
```

g_id	customer_id	room_id	check_in_date	check_out_date	payment_id	payment_method	payment_date	payment_status	amount
167	1	1	2024-08-05	2024-08-09	84	Internet Banking	2024-06-20	Completed	600.00

Câu 2: Ngăn chặn “double booking”

SQL:

```
1 CREATE OR REPLACE FUNCTION trg_prevent_double_booking()
2 RETURNS TRIGGER AS $$
3 DECLARE
4     existing_booking INT;
5 BEGIN
6     SELECT COUNT(*)
7     INTO existing_booking
8     FROM bookings
9     WHERE room_id = NEW.room_id
10    AND check_in_date <= NEW.check_out_date
11    AND check_out_date >= NEW.check_in_date;
12 IF existing_booking > 0 THEN
13     RAISE EXCEPTION 'Không thể đặt phòng vì phòng đã được đặt trong khoảng thời gian này!';
14 END IF;
15 RETURN NEW;
16 END;
17 $$ LANGUAGE plpgsql;
18 CREATE TRIGGER prevent_double_booking
19 BEFORE INSERT ON bookings
20 FOR EACH ROW
21 EXECUTE FUNCTION trg_prevent_double_booking();
```



Kết quả:

```
23 INSERT INTO bookings (customer_id, hotel_id, room_id, check_in_date, check_out_date, total_price) VALUES
24 (1, 1, 1, '2024-05-05', '2024-08-08', 450.00);
```

Data Output Messages Notifications

ERROR: Không thể đặt phòng vì phòng đã được đặt trong khoảng thời gian này!
CONTEXT: PL/pgSQL function trg_prevent_double_booking() line 12 at RAISE

SQL state: P0001

- Trong khoảng thời gian 5/8/2024 đến 7/8/2024 đã có người đặt phòng

84	167	1	1	1	2024-08-05	2024-08-07	300.00
----	-----	---	---	---	------------	------------	--------

Câu 3: Khi bảng bookings được thêm bản ghi, thêm tương ứng bản ghi ở bảng payments với yêu cầu: payment_method = NULL, payment_date = NULL, payment_status = 'In Progress' và payments.amount = bookings.total_price

SQL:

```
1 CREATE OR REPLACE FUNCTION trg_after_insert_bookings()
2 RETURNS TRIGGER AS $$
3 BEGIN
4     INSERT INTO payments (booking_id, payment_method, payment_date, payment_status, amount)
5     VALUES (NEW.booking_id, NULL, NULL, 'In Progress', NEW.total_price);
6     RETURN NEW;
7 END;
8 $$ LANGUAGE plpgsql;
9 CREATE TRIGGER after_insert_bookings
10 AFTER INSERT ON bookings
11 FOR EACH ROW
12 EXECUTE FUNCTION trg_after_insert_bookings();
```

Kết quả:

INSERT INTO bookings (customer_id, hotel_id, room_id, check_in_date, check_out_date, total_price) VALUES

(1, 1, 1, '2024-05-05', '2024-08-07', 300.00);

SELECT * FROM payments WHERE payment_id=84;

	payment_id [PK] integer	booking_id integer	payment_method character varying (50)	payment_date date	payment_status character varying (50)	amount numeric (10,2)
1	84	167	[null]	[null]	In Progress	300.00

Câu 4: Khi update payment_date, cập nhật payment_status = 'Completed'

SQL:



```

1 CREATE OR REPLACE FUNCTION trg_update_payment_status()
2 RETURNS TRIGGER AS $$
3 BEGIN
4     IF NEW.payment_date IS DISTINCT FROM OLD.payment_date AND NEW.payment_method IS DISTINCT FROM OLD.payment_method THEN
5         UPDATE payments
6             SET payment_status = 'Completed'
7             WHERE payment_id = NEW.payment_id;
8     END IF;
9     RETURN NEW;
10 END;
11 $$ LANGUAGE plpgsql;
12 CREATE TRIGGER update_payment_status
13 BEFORE UPDATE OF payment_date, payment_method ON payments
14 FOR EACH ROW
15 EXECUTE FUNCTION trg_update_payment_status();

```

Kết quả:

UPDATE payments SET payment_method='Internet Banking', payment_date = '2024-06-20' WHERE payment_id = 84;

SELECT * FROM payments WHERE payment_id = 84;

	payment_id [PK] integer	booking_id integer	payment_method character varying (50)	payment_date date	payment_status character varying (50)	amount numeric (10,2)
1	84	167	Internet Banking	2024-06-20	Completed	300.00

2.5. Index

Câu 1: tăng tốc độ truy vấn khi tìm kiếm các khách sạn theo thành phố

SQL:

```
2 CREATE INDEX idx_hotels_city ON hotels (city);
```

Kết quả:

- Chưa có Index:

Query complete 00:00:00.112

- Có Index:

Query complete 00:00:00.079

Câu 2: cải thiện hiệu suất khi tìm kiếm các đơn đặt phòng của một khách hàng cụ thể

SQL:

```
2 CREATE INDEX idx_bookings_customer_id ON bookings (customer_id);
```

Kết quả:

- Chưa có Index:

Query complete 00:00:00.120

- Có Index:

Query complete 00:00:00.067

Câu 3: tìm kiếm các đặt phòng trong một khách sạn theo khoảng thời gian nhất định

SQL:

```
5 CREATE INDEX idx_bookings_hotel_id_check_in_date ON bookings (hotel_id, check_in_date);
```

Kết quả:

- Chưa có Index:

Query complete 00:00:00.090

- Có Index:

Query complete 00:00:00.065

IV. Kết luận

1. Hoàn thành thiết kế cơ sở dữ liệu cho Website đặt phòng khách sạn, đảm bảo được các yếu tố:

- Xây dựng thành công cơ sở dữ liệu dựa trên các yếu tố thực
- Tạo dữ liệu trên cơ sở thực tế
- Thực hiện truy vấn dựa trên các tình huống trong thực tế
- Hoàn thiện báo cáo thu hoạch

2. Kiến thức thu được

- Vận dụng các kiến thức thực hành cơ sở dữ liệu đã học
- Biết cách dữ liệu hóa các thực thể, xây dựng được các mối liên kết
- Tối ưu các câu lệnh SQL

3. Vai trò của các thành viên

- Nguyễn Mạnh Tùng: Xây dựng cơ sở dữ liệu, viết truy vấn, tối ưu câu lệnh, viết báo cáo, lập trình giao diện Website demo.
- Tạ Hồng Phúc: Xây dựng cơ sở dữ liệu, viết truy vấn, tối ưu câu lệnh, lập trình giao diện, viết API tĩnh, xây dựng back-end cho Website demo.
- Bùi Quang Hưng: Xây dựng cơ sở dữ liệu, viết truy vấn, lập trình giao diện + back-end cho Website demo.

3. Source Code:

- File SQL đã bao gồm trong folder nộp bài tập nhóm
- Link WEBSITE demo sản phẩm: [Github andrew-taphuc Hotel-Reservation-Hust](https://github.com/andrew-taphuc/Hotel-Reservation-Hust)