

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



BÁO CÁO BÀI TẬP CUỐI KÌ
MÔN THỰC HÀNH CƠ SỞ DỮ LIỆU – IT3290 KÌ 2023.2

Giảng viên hướng dẫn: Nguyễn Hữu Đức

Sinh viên thực hiện: Nguyễn Xuân Khuê – 20225643

Nguyễn Hoàng Tùng – 20225948

Nguyễn Hồng Quân – 20225908

Hà Nội, ngày 27 tháng 06 năm 2024

MỤC LỤC

I. MÔ TẢ BÀI TOÁN	4
1.1. Đề tài: CƠ SỞ DỮ LIỆU QUẢN LÝ KHÁCH SẠN	4
1.2. Bối cảnh	4
1.3. Mô tả cơ sở dữ liệu	4
II. MÔ HÌNH QUAN HỆ	5
2.1. Sơ đồ thực thể liên kết	5
2.2. Sơ đồ quan hệ	6
III. THIẾT KẾ CƠ SỞ DỮ LIỆU	6
3.1. Thiết kế các quan hệ	6
• Bảng khách hàng	6
• Bảng phòng	7
• Bảng đặt phòng	7
• Bảng nhân viên	7
• Bảng dịch vụ	8
• Bảng đặt phòng dịch vụ	8
• Bảng phụ trách	9
IV. CẤP QUYỀN	9
4.1. Khách hàng	9
4.2. Nhân viên	9
4.3. Admin	9
V. CÁC TRUY VẤN	9
5.1. Truy vấn đơn giản	9
• Lấy tất cả thông tin từ bảng customers	9
• Lấy danh sách tên và email của tất cả khách hàng	10
• Kiểm tra tổng số lượng dịch vụ đã sử dụng trong mỗi booking (kết quả từ trigger)	10
• Kiểm tra lương của từng nhân viên (kết quả từ trigger cập nhật lương)	11
5.2. Truy vấn điều kiện	11
• Lấy thông tin của khách hàng có customer_id bằng 1	11
• Lấy tất cả các phòng có giá trên 100	11
5.3. Truy vấn với các hàm tính toán	12
• Tính tổng số lượng dịch vụ được sử dụng	12

• Tính tổng doanh thu từ các dịch vụ.....	12
5.4. Truy vấn với Group By.....	12
• Tính tổng doanh thu theo từng loại dịch vụ.....	12
• Tính số lượng phòng đã được đặt bởi từng khách hàng.....	13
5.5. Truy vấn với Join	14
• Lấy thông tin về tất cả các booking cùng với thông tin khách hàng.....	14
• Lấy thông tin về các dịch vụ được sử dụng trong mỗi booking	14
5.6. Truy vấn con	15
• Lấy thông tin khách hàng có tổng giá trị đặt phòng cao nhất.....	15
• Lấy thông tin về các dịch vụ được sử dụng nhiều nhất.....	16
VI. TRIGGER & FUNCTION.....	16
6.1. Trigger để cập nhật tổng số lượng dịch vụ đã sử dụng.....	16
6.2. Trigger để tính toán và cập nhật lương của nhân viên	18
6.3. Trigger kiểm tra mỗi khi có đặt phòng	19
VII. TỔNG KẾT.....	21

I. MÔ TẢ BÀI TOÁN

1.1. Đề tài: CƠ SỞ DỮ LIỆU QUẢN LÝ KHÁCH SẠN

1.2. Bối cảnh

Bối cảnh xây dựng cơ sở dữ liệu quản lý khách sạn bao gồm việc quản lý thông tin khách hàng, đặt phòng, dịch vụ và thanh toán. Hệ thống cần phải theo dõi thông tin chi tiết của từng khách hàng, lịch sử lưu trú và yêu cầu đặc biệt. Cơ sở dữ liệu cũng phải hỗ trợ quản lý phòng, tình trạng phòng trống, và lịch trình dọn phòng. Ngoài ra, hệ thống cần tích hợp với các dịch vụ phụ trợ như. Cuối cùng, cơ sở dữ liệu phải đảm bảo an ninh thông tin và tuân thủ các quy định pháp luật về bảo mật dữ liệu.

1.3. Mô tả cơ sở dữ liệu

Cơ sở dữ liệu quản lý khách sạn được thiết kế để tổ chức và quản lý các thông tin liên quan đến hoạt động của một khách sạn. Cơ sở dữ liệu này thường bao gồm các bảng và mối quan hệ giữa chúng để quản lý hiệu quả các khía cạnh sau:

Thông tin khách hàng: Bảng này lưu trữ các thông tin cá nhân của khách hàng, bao gồm tên, địa chỉ, số điện thoại, email, và thông tin nhận dạng khác. Mỗi khách hàng sẽ có một mã định danh duy nhất.

Đặt phòng: Bảng này chứa các thông tin về đặt phòng của khách hàng, bao gồm mã đặt phòng, mã khách hàng, mã phòng, ngày nhận phòng, ngày trả phòng, và tình trạng đặt phòng (xác nhận, chờ, hủy bỏ).

Phòng: Bảng này quản lý các thông tin về phòng, bao gồm mã phòng, loại phòng, tình trạng phòng (có sẵn, đã đặt, đang bảo trì), và giá phòng.

Dịch vụ: Bảng này lưu trữ thông tin về các dịch vụ mà khách sạn cung cấp như nhà hàng, spa, giặt ủi, và dịch vụ đưa đón. Mỗi dịch vụ sẽ có một mã dịch vụ, mô tả dịch vụ, và giá cả.

Hóa đơn: Bảng này quản lý các hóa đơn thanh toán của khách hàng, bao gồm mã hóa đơn, mã khách hàng, mã đặt phòng, danh sách các dịch vụ đã sử dụng, tổng số tiền, và tình trạng thanh toán.

Nhân viên: Bảng này chứa thông tin về nhân viên của khách sạn, bao gồm mã nhân viên, tên, chức vụ, bộ phận, và thông tin liên lạc.

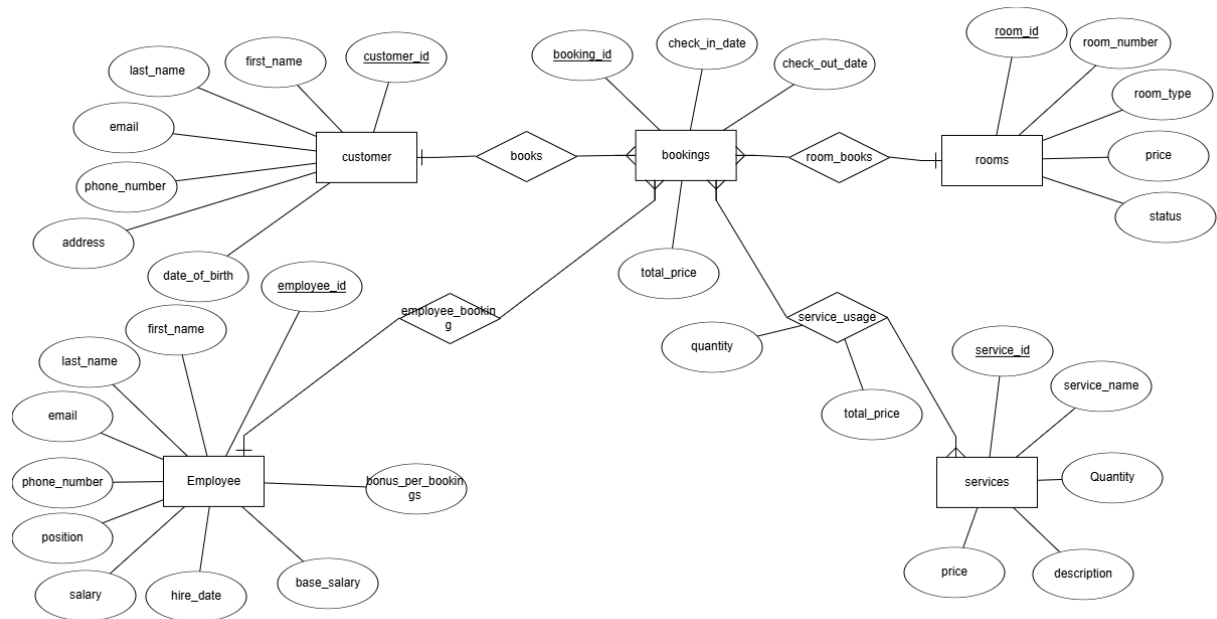
Lịch sử lưu trú: Bảng này theo dõi lịch sử lưu trú của khách hàng, bao gồm mã khách hàng, mã phòng, ngày nhận phòng, ngày trả phòng, và đánh giá của khách hàng.

Các bảng này được liên kết với nhau thông qua các khóa ngoại, giúp dễ dàng truy xuất và quản lý thông tin. Ví dụ, bảng Đặt phòng sẽ liên kết với bảng Khách hàng và

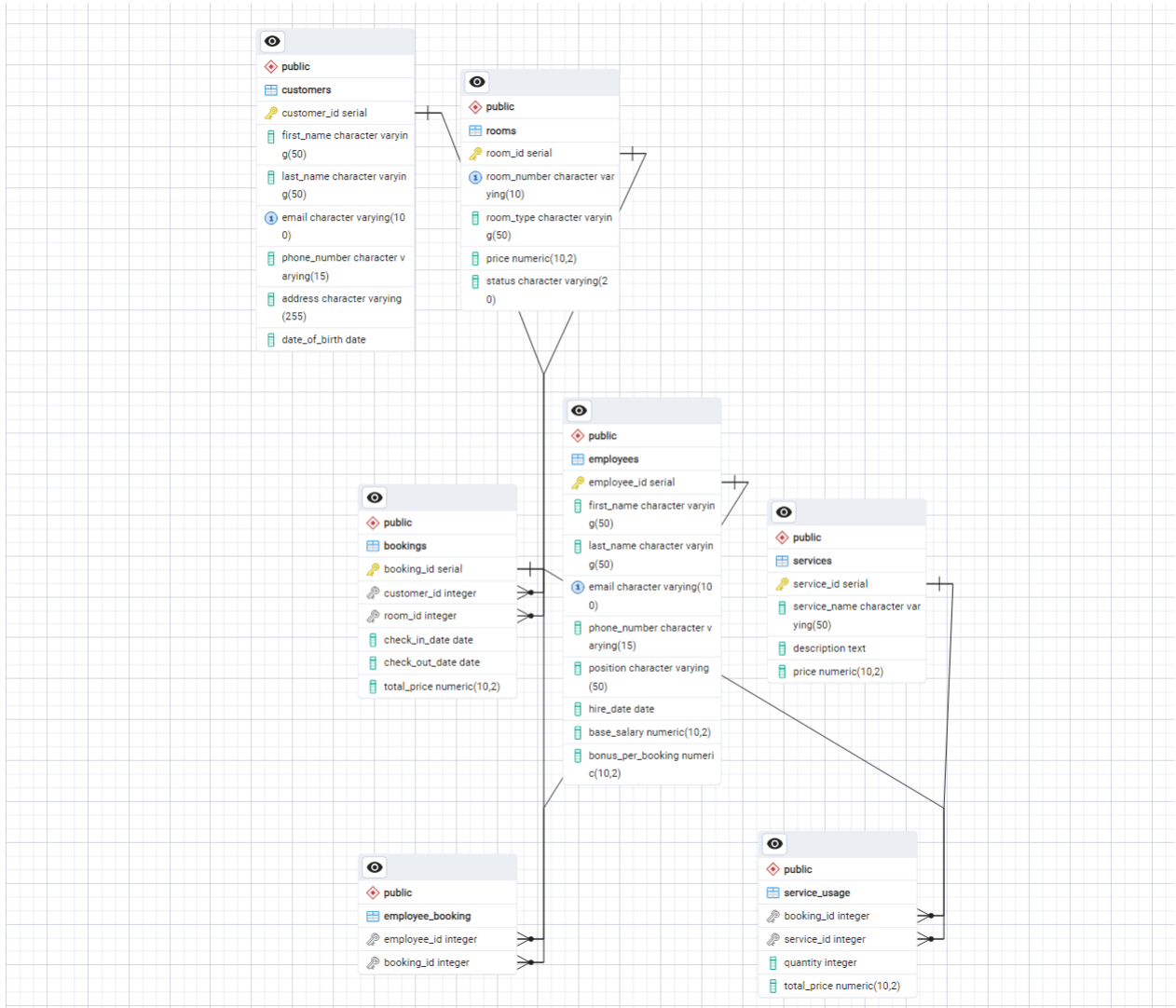
bảng Phòng thông qua mã khách hàng và mã phòng, giúp theo dõi thông tin đặt phòng và tình trạng phòng một cách hiệu quả.

II. MÔ HÌNH QUAN HỆ

2.1. Sơ đồ thực thể liên kết



2.2. Sơ đồ quan hệ



III. THIẾT KẾ CƠ SỞ DỮ LIỆU

3.1. Thiết kế các quan hệ

- **Bảng khách hàng**

CREATE TABLE customers (

customer_id SERIAL PRIMARY KEY,

first_name VARCHAR(50) NOT NULL,

last_name VARCHAR(50) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

```
phone_number VARCHAR(15) NOT NULL,  
address VARCHAR(255),  
date_of_birth DATE  
);
```

- **Bảng phòng**

```
CREATE TABLE rooms (  
    room_id SERIAL PRIMARY KEY,  
    room_number VARCHAR(10) UNIQUE NOT NULL,  
    room_type VARCHAR(50) NOT NULL,  
    price NUMERIC(10, 2) NOT NULL,  
    status VARCHAR(20) NOT NULL  
);
```

- **Bảng đặt phòng**

```
CREATE TABLE bookings (  
    booking_id SERIAL PRIMARY KEY,  
    customer_id INTEGER REFERENCES customers(customer_id),  
    room_id INTEGER REFERENCES rooms(room_id),  
    check_in_date DATE NOT NULL,  
    check_out_date DATE NOT NULL,  
    total_price NUMERIC(10, 2) NOT NULL  
);
```

- **Bảng nhân viên**

```
CREATE TABLE employees (  
    employee_id SERIAL PRIMARY KEY,  
    first_name VARCHAR(50) NOT NULL,
```

```
last_name VARCHAR(50) NOT NULL,  
email VARCHAR(100) UNIQUE NOT NULL,  
phone_number VARCHAR(15) NOT NULL,  
position VARCHAR(50),  
hire_date DATE,  
base_salary NUMERIC(10, 2),  
bonus_per_booking NUMERIC(10, 2)  
);
```

- **Bảng dịch vụ**

```
CREATE TABLE services (  
    service_id SERIAL PRIMARY KEY,  
    service_name VARCHAR(50) NOT NULL,  
    description TEXT,  
    price NUMERIC(10, 2) NOT NULL  
    quantity INTEGER NOT NULL DEFAULT 0  
);
```

- **Bảng đặt phòng dịch vụ**

```
CREATE TABLE service_usage (  
    booking_id INTEGER REFERENCES bookings(booking_id),  
    service_id INTEGER REFERENCES services(service_id),  
    quantity INTEGER NOT NULL,  
    total_price NUMERIC(10, 2) NOT NULL  
);
```


- **Bảng phụ trách**

```
CREATE TABLE employee_booking (  
    employee_id INTEGER REFERENCES employees(employee_id),  
    booking_id INTEGER REFERENCES bookings(booking_id)  
);
```

IV. CẤP QUYỀN

4.1. Khách hàng

- Tạo đặt phòng
- Xem thông tin đặt phòng của mình
- Hủy đặt phòng của mình

4.2. Nhân viên

- Quản lý đặt phòng
- Quản lý thông tin khách hàng
- Hỗ trợ khách hàng trong việc đặt phòng và giải đáp thắc mắc
- Cập nhật trạng thái phòng

4.3. Admin

- Quản lý tất cả khách hàng
- Quản lý tất cả đặt phòng
- Quản lý thông tin khách sạn (phòng, giá,...)
- Quản lý nhân viên
- Quản lý thống kê và báo cáo

V. CÁC TRUY VẤN

5.1. Truy vấn đơn giản

- **Lấy tất cả thông tin từ bảng customers**

```
SELECT * FROM customers;
```

Data Output Messages Notifications							
	customer_id [PK] integer	first_name character varying (50)	last_name character varying (50)	email character varying (100)	phone_number character varying (15)	address character varying (255)	date_of_birth date
1	1	John	Doe	john.doe@example.com	123456789	123 Elm St	1990-01-01
2	2	Jane	Smith	jane.smith@example.com	987654321	456 Oak St	1985-02-02
3	3	Alice	Johnson	alice.johnson@example.com	555555555	789 Pine St	1978-03-03
4	4	Bob	Brown	bob.brown@example.com	111222333	101 Maple St	1980-04-04
5	5	Charlie	Davis	charlie.davis@example.com	444666777	202 Cedar St	1992-05-05
6	6	Emily	Wilson	emily.wilson@example.com	666777888	303 Birch St	1987-06-06
7	7	Frank	Moore	frank.moore@example.com	777888999	404 Walnut St	1983-07-07
8	8	Grace	Taylor	grace.taylor@example.com	888999000	505 Spruce St	1988-08-08
9	9	Hannah	Anderson	hannah.anderson@example.com	999000111	606 Willow St	1975-09-09
10	10	Ian	Thomas	ian.thomas@example.com	000111222	707 Cherry St	1992-10-10

- **Lấy danh sách tên và email của tất cả khách hàng**

SELECT first_name, last_name, email FROM customers;

	first_name character varying (50)	last_name character varying (50)	email character varying (100)
1	John	Doe	john.doe@example.com
2	Jane	Smith	jane.smith@example.com
3	Alice	Johnson	alice.johnson@example.com
4	Bob	Brown	bob.brown@example.com
5	Charlie	Davis	charlie.davis@example.com
6	Emily	Wilson	emily.wilson@example.com
7	Frank	Moore	frank.moore@example.com

- **Kiểm tra tổng số lượng dịch vụ đã sử dụng trong mỗi booking (kết quả từ trigger)**

SELECT booking_id, total_service_usage

FROM bookings;

	booking_id [PK] integer	total_service_usage integer
1	1	2
2	2	1
3	3	1
4	4	3
5	5	2

- Kiểm tra lương của từng nhân viên (kết quả từ trigger cập nhật lương)

SELECT employee_id, first_name, last_name, salary

FROM employees;

	employee_id [PK] integer	first_name character varying (50)	last_name character varying (50)	salary numeric
1	1	Alice	Walker	5150.00
2	2	Bob	Harris	2050.00
3	3	Carol	Moore	1530.00
4	4	David	Taylor	5200.00
5	5	Eve	Johnson	2260.00

5.2. Truy vấn điều kiện

- Lấy thông tin của khách hàng có customer_id bằng 1

SELECT * FROM customers WHERE customer_id = 1;

	customer_id [PK] integer	first_name character varying (50)	last_name character varying (50)	email character varying (100)	phone_number character varying (15)	address character varying (255)	date_of_birth date
1	1	John	Doe	john.doe@example.com	123456789	123 Elm St	1990-01-01

- Lấy tất cả các phòng có giá trên 100

SELECT * FROM rooms WHERE price > 100;

	room_id [PK] integer	room_number character varying (10)	room_type character varying (50)	price numeric (10,2)	status character varying (20)
1	2	102	Double	150.00	Available
2	3	103	Suite	350.00	Available
3	5	105	Double	150.00	Booked
4	6	106	Suite	350.00	Available
5	8	108	Double	150.00	Booked
6	9	109	Suite	350.00	Booked
7	12	202	Double	150.00	Available

5.3. Truy vấn với các hàm tính toán

- Tính tổng số lượng dịch vụ được sử dụng

SELECT SUM(quantity) FROM service_usage;

	sum bigint
1	34

- Tính tổng doanh thu từ các dịch vụ

SELECT SUM(total_price) FROM service_usage;

	sum numeric
1	1265.00

5.4. Truy vấn với Group By

- Tính tổng doanh thu theo từng loại dịch vụ

SELECT service_id, SUM(total_price) AS total_revenue
FROM service_usage
GROUP BY service_id;

	service_id integer	total_revenue numeric
1	9	280.00
2	3	100.00
3	5	30.00
4	4	60.00
5	10	100.00
6	6	150.00
7	2	75.00

- **Tính số lượng phòng đã được đặt bởi từng khách hàng**

SELECT customer_id, COUNT(*) AS total_bookings

FROM bookings

GROUP BY customer_id;

	customer_id integer	total_bookings bigint
1	11	1
2	9	1
3	15	1
4	19	1
5	3	1
6	17	1
7	5	1

* **Đánh giá hiệu năng truy vấn:**

	QUERY PLAN text
1	HashAggregate (cost=1.31..1.52 rows=21 width=12)
2	Group Key: customer_id
3	Batches: 1 Memory Usage: 24kB
4	-> Seq Scan on bookings (cost=0.00..1.21 rows=21)
5	Planning Time: 0.209 ms
6	Execution Time: 0.040 ms

20 bản ghi: 30 ms

1000 bản ghi: 47 ms

1000 bản ghi có index: 35ms

CREATE INDEX idx_customer_id ON bookings (customer_id)

5.5. Truy vấn với Join

- **Lấy thông tin về tất cả các booking cùng với thông tin khách hàng**

SELECT b.booking_id, b.check_in_date, b.check_out_date, c.first_name, c.last_name

FROM bookings b

JOIN customers c ON b.customer_id = c.customer_id;

	booking_id integer	check_in_date date	check_out_date date	first_name character varying (50)	last_name character varying (50)
1	2	2024-06-02	2024-06-06	Jane	Smith
2	3	2024-06-03	2024-06-07	Alice	Johnson
3	4	2024-06-04	2024-06-08	Bob	Brown
4	5	2024-06-05	2024-06-09	Charlie	Davis
5	6	2024-07-10	2024-07-15	Emily	Wilson
6	7	2024-07-11	2024-07-16	Frank	Moore
7	8	2024-07-12	2024-07-17	Grace	Taylor

- * **Đánh giá hiệu năng truy vấn**

	QUERY PLAN text
1	Hash Join (cost=1.47..12.65 rows=21 width=248) (actual tin
2	Hash Cond: (c.customer_id = b.customer_id)
3	-> Seq Scan on customers c (cost=0.00..10.70 rows=70 wi
4	-> Hash (cost=1.21..1.21 rows=21 width=16) (actual time=
5	Buckets: 1024 Batches: 1 Memory Usage: 9kB
6	-> Seq Scan on bookings b (cost=0.00..1.21 rows=21 w
7	Planning Time: 2.262 ms
8	Execution Time: 0.202 ms

*

20 bản ghi: 68 ms

1000 bản ghi: 80 ms

1000 bản ghi có index: 75 ms

CREATE INDEX idx_booking_id ON bookings (booking_id)

- **Lấy thông tin về các dịch vụ được sử dụng trong mỗi booking**

SELECT b.booking_id, s.service_name, su.quantity, su.total_price

FROM service_usage su

JOIN services s ON su.service_id = s.service_id

JOIN bookings b ON su.booking_id = b.booking_id;

	booking_id integer	service_name character varying (50)	quantity integer	total_price numeric (10,2)
1	1	Room Service	2	100.00
2	2	Laundry	1	30.00
3	3	Spa	1	100.00
4	4	Gym	3	60.00
5	5	Breakfast	2	30.00
6	6	Mini Bar	1	30.00

* Đánh giá hiệu năng truy vấn

	QUERY PLAN text
1	Hash Join (cost=20.70..51.55 rows=1360 width=142) (actual tim
2	Hash Cond: (su.booking_id = b.booking_id)
3	-> Hash Join (cost=19.23..46.43 rows=1360 width=142) (actua
4	Hash Cond: (su.service_id = s.service_id)
5	-> Seq Scan on service_usage su (cost=0.00..23.60 rows=1:
6	-> Hash (cost=14.10..14.10 rows=410 width=122) (actual til
7	Buckets: 1024 Batches: 1 Memory Usage: 9kB
8	-> Seq Scan on services s (cost=0.00..14.10 rows=410 v
9	-> Hash (cost=1.21..1.21 rows=21 width=4) (actual time=0.010
10	Buckets: 1024 Batches: 1 Memory Usage: 9kB
11	-> Seq Scan on bookings b (cost=0.00..1.21 rows=21 width:
12	Planning Time: 1.462 ms
13	Execution Time: 0.284 ms

*

20 bản ghi: 70 ms

1000 bản ghi: 78 ms

1000 bản ghi có index: 72 ms

CREATE INDEX idx_booking_id ON service_usage (booking_id)

5.6. Truy vấn con

- Lấy thông tin khách hàng có tổng giá trị đặt phòng cao nhất

SELECT * FROM customers

WHERE customer_id = (

SELECT customer_id

FROM bookings

GROUP BY customer_id

ORDER BY SUM(total_price) DESC

LIMIT 1

);

	customer_id [PK] integer	first_name character varying (50)	last_name character varying (50)	email character varying (100)	phone_number character varying (15)	address character varying (255)	date_of_birth date
1	9	Hannah	Anderson	hannah.anderson@example.com	999000111	606 Willow St	1975-09-09

- **Lấy thông tin về các dịch vụ được sử dụng nhiều nhất**

SELECT * FROM services

WHERE service_id = (

SELECT service_id

FROM service_usage

GROUP BY service_id

ORDER BY SUM(quantity) DESC

LIMIT 1

);

	service_id [PK] integer	service_name character varying (50)	description text	price numeric (10,2)
1	10	Breakfast Buffet	Full breakfast buffet	20.00

VI. TRIGGER & FUNCTION

6.1. Trigger để cập nhật tổng số lượng dịch vụ đã sử dụng

Trigger để cập nhật tổng số lượng dịch vụ đã sử dụng sau mỗi lần thêm hoặc xóa dịch vụ trong bảng service_usage:

- Kiểm tra số lượng dịch vụ khi thêm mới
- Trả số lượng dịch vụ khi xóa khỏi service_usage
- Cập nhật tổng số lượng dịch vụ đã sử dụng trong bookings

Trigger:

CREATE OR REPLACE FUNCTION update_total_service_usage()

RETURNS TRIGGER AS

\$\$

BEGIN

-- Kiem tra so luong con lai cua dich vu khi them moi

IF TG_OP = 'INSERT' THEN

IF NEW.quantity > (SELECT quantity FROM services WHERE service_id =
NEW.service_id) THEN

RAISE EXCEPTION 'Not enough quantity for service %', NEW.service_id;
END IF;

-- Cap nhat so luong dich vu con lai

UPDATE services

SET quantity = quantity - NEW.quantity

WHERE service_id = NEW.service_id;

-- Hoan tra so luong dich vu khi xoa khoi service_usage

ELSIF TG_OP = 'DELETE' THEN

UPDATE services

SET quantity = quantity + OLD.quantity

WHERE service_id = OLD.service_id;

END IF;

-- Cap nhat tong so luong dich vu da su dung trong bookings

UPDATE bookings

SET total_service_usage = (SELECT SUM(quantity) FROM service_usage
WHERE booking_id = NEW.booking_id)

```
WHERE booking_id = NEW.booking_id;
```

```
RETURN NEW;
```

```
END;
```

```
$$
```

```
LANGUAGE plpgsql;
```

```
CREATE TRIGGER update_service_usage_trigger
```

```
AFTER INSERT OR DELETE ON service_usage
```

```
FOR EACH ROW
```

```
EXECUTE FUNCTION update_total_service_usage();
```

Thực hiện insert lại các data trước đó ở service_usage rồi vào bookings check số dịch vụ ứng với mỗi booking_id

6.2. Trigger để tính toán và cập nhật lương của nhân viên

Trigger để tính toán và cập nhật lương của nhân viên dựa trên số lượng đơn đặt phòng mà nhân viên đó phụ trách. Trigger này sẽ được kích hoạt sau khi thêm một bản ghi trong bảng employee_booking.

Lương của 1 nhân viên sẽ được tính bằng lương cố định cộng với bonus ứng với từng đơn booking mà nhân viên đó phụ trách

Trigger + Function:

```
CREATE OR REPLACE FUNCTION update_employee_salary()
```

```
RETURNS TRIGGER AS
```

```
$$
```

```
BEGIN
```

```
    UPDATE employees
```

```
    SET salary = base_salary + (SELECT COUNT(*) * bonus_per_booking FROM  
employee_booking WHERE employee_id = NEW.employee_id)
```

```

WHERE employee_id = NEW.employee_id;
RETURN NEW;
END;
$$
LANGUAGE plpgsql;
CREATE TRIGGER update_salary_trigger
AFTER INSERT OR DELETE ON employee_booking
FOR EACH ROW
EXECUTE FUNCTION update_employee_salary();

```

Thực hiện insert/delete trong bảng employee_booking rồi sang check output(salary) bên bảng employees

6.3. Trigger kiểm tra mỗi khi có đặt phòng

```

-- FUNCTION: public.booking_trigger_function()

-- DROP FUNCTION IF EXISTS public.booking_trigger_function();

CREATE OR REPLACE FUNCTION public.booking_trigger_function()
    RETURNS trigger
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE NOT LEAKPROOF
AS $BODY$
BEGIN
    -- Kiểm tra trạng thái của đặt phòng
    IF NEW.status = 'available' THEN

```

```

-- Các xử lý khi đặt phòng thành công ở đây
-- Ví dụ: có thể cập nhật trạng thái sang 'booked' và lưu các thông tin khác
UPDATE bookings
SET status = 'booked'
WHERE booking_id = NEW.booking_id;
ELSE
-- Xử lý khi đặt phòng không thành công do trạng thái không phù hợp
RAISE EXCEPTION 'Cannot book a room with status %', NEW.status;
END IF;

RETURN NEW;
END;
$BODY$;

ALTER FUNCTION public.booking_trigger_function()
OWNER TO postgres;

```

VII. TỔNG KẾT

Nhận xét của giảng viên

Sinh viên thực hiện

(kí và ghi rõ họ tên)

Nguyễn Xuân Khuê

Nguyễn Hoàng Tùng

Nguyễn Hồng Quân