



HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.



Xây dựng CSDL Quản lý khách sạn

- Giảng viên: Trần Văn Đặng
- Nhóm thực hiện:
 - Lê Thành An 20235631
 - Lê Văn Thành An 20236018
 - Trần Đức Nam Anh 20235655



ONE LOVE. ONE FUTURE.

MỤC LỤC

01 Mô tả Bài toán thực tế

02 Phân tích nghiệp vụ - Các chức năng chính

03 Mô hình thực thể liên kết

04 Triển khai hệ thống

05 Viết truy vấn cá nhân



ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

3

Nội dung

1. Mô tả bài toán thực tế

Trong bối cảnh ngành du lịch và khách sạn ngày càng phát triển, việc quản lý số lượng lớn khách hàng, phòng ở, dịch vụ, cũng như các nghiệp vụ đặt phòng, thanh toán, khuyến mãi... đặt ra yêu cầu cao về hệ thống thông tin quản lý. Dự án này tập trung vào việc phân tích, thiết kế và xây dựng cơ sở dữ liệu quản lý khách sạn, đảm bảo các chức năng lưu trú, truy xuất, thống kê và bảo mật dữ liệu. Hệ thống hướng tới tính linh hoạt, mở rộng và đáp ứng tốt nhu cầu thực tế của các mô hình khách sạn hiện đại.



Nhóm 2 2.Phân tích nghiệp vụ - Các chức năng chính

• 1.Khách hàng đặt phòng

◦ Đặt phòng online

- Khách hàng đăng nhập hoặc đăng ký tài khoản để vào hệ thống.
- Hệ thống hiển thị các loại phòng còn trống, số lượng, giá cả, tiện ích nổi bật.
- Khách hàng xem được các ưu đãi hiện có và hạng thành viên của mình để nhận thêm khuyến mãi.
- Chương trình khuyến mãi được quản lý, cập nhật theo từng giai đoạn và có thể áp dụng theo loại phòng, thời gian ở, hạng khách.
- Khách hàng có thể sử dụng bộ lọc để tìm phòng phù hợp với nhu cầu.
- Khi thanh toán, hóa đơn ghi rõ đầy đủ thông tin và được lưu vào lịch sử thanh toán.

◦ Hủy phòng online

- Khách hàng vào hóa đơn và yêu cầu hủy phòng đã đặt.
- Hệ thống cũng cập nhật hóa đơn mới, cập nhật tình trạng phòng đã được hủy.



Nội dung

2.Phân tích nghiệp vụ - Các chức năng chính

• 2. Quản lý đặt phòng

- Hệ thống quản lý lịch đặt phòng, bao gồm các trạng thái: đã đặt, đã xác nhận, đã nhận phòng, đã huỷ, đã trả phòng,..
- Tự động kiểm tra và thông báo khi sắp hết phòng hoặc có phòng trống đột xuất
- Hỗ trợ đặt theo loại phòng, số người, dịch vụ kèm theo (ăn sáng, spa, tour du lịch)



• 3. Quản lý khách hàng

- Lưu trữ đầy đủ thông tin khách hàng: CCCD/Passport, tên, số điện thoại, email, quốc tịch, lịch sử đặt phòng.
- Quản lý các yêu cầu đặc biệt khi đặt phòng (ăn chay, phòng gần cửa sổ, nội em bé...).
- Ghi nhận lịch sử tương tác, bao gồm các yêu cầu đặc biệt, khiếu nại, phản hồi.
- Sau kỳ nghỉ, khách hàng có thể đánh giá phòng/dịch vụ; đánh giá được liên kết với từng đặt phòng.
- Phản hồi của khách được quản trị viên kiểm duyệt, phân tích để cải thiện dịch vụ; các phòng/dịch vụ nhiều phản hồi tiêu cực sẽ được kiểm tra.



Nội dung 2.Phân tích nghiệp vụ - Các chức năng chính



• 4. Quản lý thu chi (Hoá đơn và thanh toán)

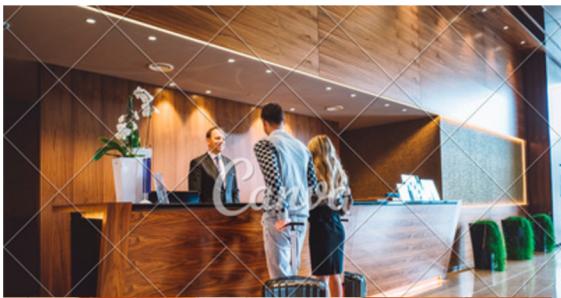
- Ghi nhận và quản lý các giao dịch thanh toán từ khách hàng (tiền mặt, thẻ, ví điện tử)
- Tự động xuất hoá đơn với chi tiết giá phòng, dịch vụ đi kèm, thuế, phụ phí (nếu có)
- Kiểm soát công nợ khách hàng (nếu có hình thức đặt trước/trả sau), Thống kê thu chi:
- Doanh thu theo ngày/tuần/tháng/năm, So sánh doanh thu theo từng loại phòng, dịch vụ. Tổng hợp chi phí vận hành khách sạn, Lợi nhuận sau khi trừ chi phí

• 5. Quản lý kho vật tư

- Nhân viên vật tư thường xuyên kiểm tra số lượng và tình trạng thiết bị dự phòng trong kho.
- Thiết bị hỏng hoặc sắp hết phải được báo cáo, xử lý kịp thời và lên kế hoạch đặt hàng bổ sung.
- Khi khách trả phòng, kiểm tra và thay thế vật dụng hỏng; cập nhật giảm số lượng kho và ghi nhận người thực hiện.
- Khi nhập hàng, nhân viên ghi rõ số lượng, nhà cung cấp, giá, cập nhật vào hệ thống cùng thông tin người phụ trách và thời gian.



Nhóm 2.Phân tích nghiệp vụ - Các chức năng chính



• 6. Quản lý check in / check out

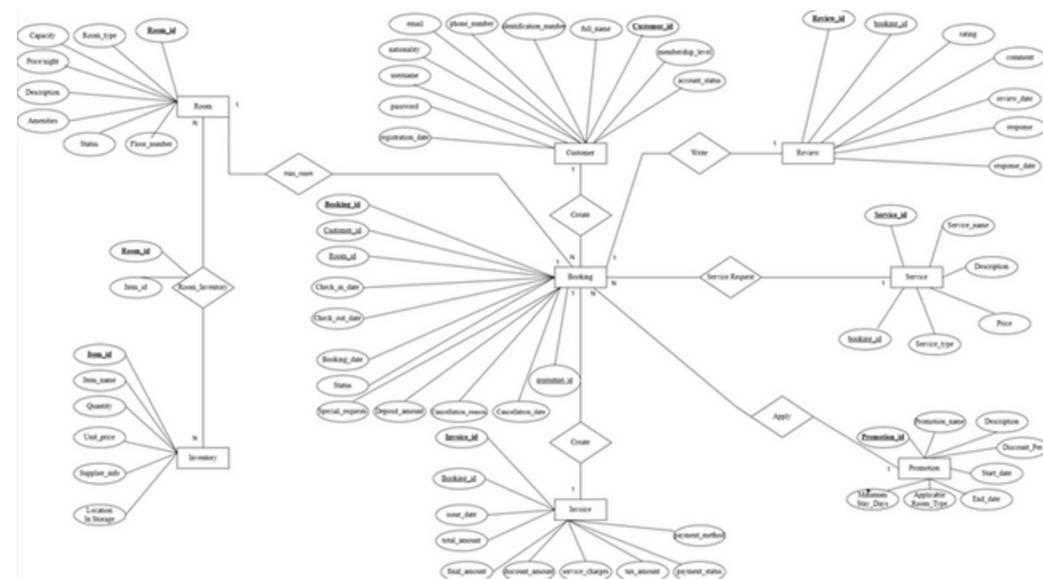
- Khi nhận phòng, khách chỉ cần xuất hóa đơn điện tử để quét mã, nhận chìa khóa và lên phòng; hệ thống sẽ cập nhật tình trạng phòng.
- Khi trả phòng, khách trả chìa khóa; nhân viên kiểm tra tình trạng phòng, vật dụng, các khoản phí dịch vụ, cập nhật hóa đơn và chuyển trạng thái phòng sang đã checkout, đồng thời tích điểm cho khách.

• 7. Quản lý tiện ích dịch vụ

- Khách hàng đang trú tại khách sạn có thể yêu cầu dịch vụ. Sau đó, thông tin phòng, thông tin dịch vụ mà khách hàng chọn sẽ được gửi yêu cầu đến hệ thống, hệ thống sẽ ghi nhận và thay đổi trong hóa đơn. Chi phí dịch vụ của khách hàng sẽ được tính vào phần phí dịch vụ của hóa đơn khi họ check out.

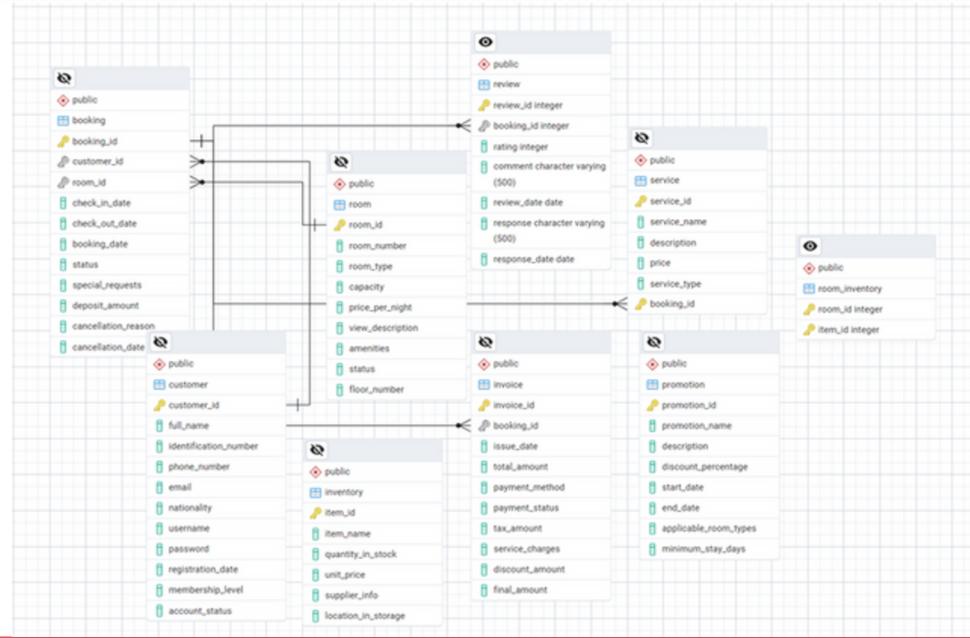
3. Mô hình thực thể liên kết

1. Sơ đồ thực thể liên kết:



3. Mô hình thực thể liên kết

2. Sơ đồ quan hệ



3. Mô hình thực thể liên kết

Các Quan Hệ:

- **Customer**(customer_id, full_name, identification_number, phone_number, email, nationality, membership_level, account_status, username, password, registration_date)
- **Room**(room_id, room_type, capacity, price_per_night, view_description, amenities, status, floor_number)
- **RoomInventory**(room_id, item_id)
- **Booking**(booking_id, customer_id, room_id, check_in_date, check_out_date, booking_date, status, special_requests, deposit_amount, cancellation_reason, cancellation_date)
- **Review**(review_id, booking_id, rating, comment, review_date, response, response_date)
- **Invoice**(invoice_id, booking_id, issue_date, total_amount, payment_method, payment_status, tax_amount, service_charges, discount_amount, final_amount)
- **Service**(service_id, booking_id, room_id, service_name, description, price, service_type)
- **Promotion**(promotion_id, promotion_name, description, discount_percentage, start_date, end_date, applicable_room_types, minimum_stay_days)
- **Inventory**(item_id, item_name, quantity_in_stock, unit_price, supplier_info, location_in_storage)

3. Mô hình thực thể liên kết

• Mô tả các thực thể, các liên kết

Thực thể A	Thực thể B	Quan hệ	Mô tả liên kết
Room	Inventory	N - N	Một phòng có
Booking	Review	1 - 1	Một đơn đặt phòng ứng với một đánh giá
Booking	Invoice	1 - 1	Một đơn đặt phòng sẽ tạo một hóa đơn
Promotion	Booking	N - 1	Một đơn đặt phòng có thể áp dụng một khuyến mãi. Một khuyến mãi có thể xuất hiện trong nhiều đơn đặt phòng
Booking	Service	1 - 1	Một đơn đặt phòng chỉ có thể yêu cầu một dịch vụ
Booking	Room	N - 1	Một đơn đặt phòng có thể đặt một phòng và một phòng có thể nằm trong nhiều đơn đặt phòng (với thời gian đặt khác nhau)
Customer	Booking	1 - N	Một khách hàng có thể có nhiều đơn đặt phòng

Nội dung bài học

4. Triển khai hệ thống

4.1. Khởi tạo bảng

4.2. Khởi tạo các trigger/function

Nội dung

4.1. Khởi tạo bảng

```
Query Query History
1 ✓ CREATE TABLE IF NOT EXISTS public.booking
2 (
3     booking_id integer NOT NULL,
4     customer_id integer NOT NULL,
5     room_id integer NOT NULL,
6     promotion_id integer,
7     check_in_date date,
8     check_out_date date,
9     booking_date date,
10    status integer,
11    special_requests VARCHAR(500) COLLATE pg_catalog."default",
12    deposit_amount numeric(10,2),
13    cancellation_reason VARCHAR(500) COLLATE pg_catalog."default",
14    cancellation_date date,
15    CONSTRAINT booking_pkey PRIMARY KEY (booking_id),
16    CONSTRAINT booking_customer_id_fkey FOREIGN KEY (customer_id)
17        REFERENCES public.customer (customer_id)
18        ON UPDATE NO ACTION
19        ON DELETE NO ACTION,
20    CONSTRAINT booking_promotion_id_fkey FOREIGN KEY (promotion_id)
21        REFERENCES public.promotion (promotion_id)
22        ON UPDATE NO ACTION
23        ON DELETE NO ACTION,
24    CONSTRAINT booking_room_id_fkey FOREIGN KEY (room_id)
25        REFERENCES public.room (room_id)
26        ON UPDATE NO ACTION
27        ON DELETE NO ACTION
28 );
```

```
Query Query History
1 ✓ CREATE TABLE IF NOT EXISTS public.invoice
2 (
3     invoice_id integer NOT NULL,
4     booking_id integer NOT NULL,
5     issue_date date,
6     total_amount numeric(10,2),
7     payment_method VARCHAR(20) COLLATE pg_catalog."default",
8     payment_status integer,
9     tax_amount numeric(10,2),
10    service_charges numeric(10,2),
11    discount_amount numeric(10,2),
12    final_amount numeric(10,2),
13    CONSTRAINT invoice_pkey PRIMARY KEY (invoice_id),
14    CONSTRAINT invoice_booking_id_fkey FOREIGN KEY (booking_id)
15        REFERENCES public.booking (booking_id)
16        ON UPDATE NO ACTION
17        ON DELETE NO ACTION
18 );
```

HUST

booking

invoice

14

Nội dung

4.1. Khởi tạo bảng

customer

inventory

HUST

15

```
Query Query History
1 ✓ CREATE TABLE IF NOT EXISTS public.customer
2 (
3     customer_id integer NOT NULL,
4     full_name VARCHAR(100) COLLATE pg_catalog."default",
5     identification_number character(12) COLLATE pg_catalog."default",
6     phone_number character(11) COLLATE pg_catalog."default",
7     email VARCHAR(100) COLLATE pg_catalog."default",
8     nationality VARCHAR(28) COLLATE pg_catalog."default",
9     username VARCHAR(100) COLLATE pg_catalog."default",
10    password VARCHAR(50) COLLATE pg_catalog."default",
11    registration_date date,
12    membership_level integer DEFAULT 0,
13    account_status integer,
14    CONSTRAINT customer_pkey PRIMARY KEY (customer_id)
15 );
Query Query History
1 ✓ CREATE TABLE IF NOT EXISTS public.inventory
2 (
3     item_id integer NOT NULL,
4     item_name VARCHAR(500) COLLATE pg_catalog."default",
5     quantity_in_stock integer,
6     unit_price numeric(10,2),
7     supplier_info VARCHAR(100) COLLATE pg_catalog."default",
8     location_in_storage VARCHAR(100) COLLATE pg_catalog."default",
9     CONSTRAINT inventory_pkey PRIMARY KEY (item_id)
10 );
```

Nội dung

4.1. Khởi tạo bảng

Query Query History

```
1 ✓ CREATE TABLE IF NOT EXISTS public.promotion
2   (
3     promotion_id integer NOT NULL,
4     promotion_name VARCHAR(100) COLLATE pg_catalog."default",
5     description VARCHAR(500) COLLATE pg_catalog."default",
6     discount_percentage numeric,
7     start_date date,
8     end_date date,
9     applicable_room_types VARCHAR(20) COLLATE pg_catalog."default",
10    minimum_stay_days integer,
11    CONSTRAINT promotion_pkey PRIMARY KEY (promotion_id)
12  );
```

promotion

Query Query History

```
1 ✓ CREATE TABLE IF NOT EXISTS public.review
2   (
3     review_id integer NOT NULL,
4     booking_id integer NOT NULL,
5     rating integer,
6     comment character varying(500) COLLATE pg_catalog."default",
7     review_date date,
8     response character varying(500) COLLATE pg_catalog."default",
9     response_date date,
10    CONSTRAINT review_pkey PRIMARY KEY (review_id),
11    CONSTRAINT review_booking_id_fkey FOREIGN KEY (booking_id)
12      REFERENCES public.booking (booking_id)
13      ON UPDATE NO ACTION
14      ON DELETE NO ACTION
15  );
```

review

Nội dung

4.1. Khởi tạo bảng

<p>Query Query History</p> <pre>1 ✓ CREATE TABLE IF NOT EXISTS public.room 2 (3 room_id integer NOT NULL, 4 room_number integer NOT NULL, 5 room_type VARCHAR(20) COLLATE pg_catalog."default", 6 capacity integer, 7 price_per_night numeric(10,2), 8 view_description VARCHAR(500) COLLATE pg_catalog."default", 9 amenities VARCHAR(100) COLLATE pg_catalog."default", 10 status integer NOT NULL, 11 floor_number integer NOT NULL, 12 CONSTRAINT room_pkey PRIMARY KEY (room_id) 13);</pre>	<p>Query Query History</p> <pre>1 ✓ CREATE TABLE IF NOT EXISTS public.service 2 (3 service_id integer NOT NULL, 4 service_name VARCHAR(100) COLLATE pg_catalog."default", 5 description VARCHAR(100) COLLATE pg_catalog."default", 6 booking_id integer , 7 price numeric(10,2), 8 service_type VARCHAR(100) COLLATE pg_catalog."default", 9 CONSTRAINT service_pkey PRIMARY KEY (service_id), 10 CONSTRAINT booking_booking_id_fkey FOREIGN KEY (booking_id) 11 REFERENCES public.booking (booking_id) MATCH SIMPLE 12 ON UPDATE NO ACTION 13 ON DELETE NO ACTION 14);</pre>
room	service
 17	

Nội dung

4.1. Khởi tạo bảng

Query Query History

```
1 ✓ CREATE TABLE IF NOT EXISTS public.room_inventory
2 (
3     room_id integer NOT NULL,
4     item_id integer NOT NULL,
5     CONSTRAINT room_inventory_pkey PRIMARY KEY (room_id, item_id)
6 );
7 |
```

room_inventory

Nội dung chương trình

4.2. Khởi tạo các trigger/function

```
CREATE OR REPLACE FUNCTION calculate_invoice_amounts()
RETURNS TRIGGER AS
$$
DECLARE
    room_price_per_night NUMERIC(10, 2);
    check_in_date DATE;
    check_out_date DATE;
    stay_days INTEGER;
    room_type_value VARCHAR(50);
    calculated_total_amount NUMERIC(10, 2);
    calculated_tax_amount NUMERIC(10, 2);
    calculated_service_charges NUMERIC(10, 2);
    calculated_discount_amount NUMERIC(10, 2);
    calculated_final_amount NUMERIC(10, 2);
    applicable_discount_rate NUMERIC(5, 2); -- phần trăm giảm giá
BEGIN
    SELECT
        r.price_per_night,
        b.check_in_date,
        b.check_out_date,
        r.room_type
    INTO
        room_price_per_night,
        check_in_date,
        check_out_date,
        room_type_value
    FROM booking b
    JOIN room r ON b.room_id = r.room_id
    WHERE b.booking_id = NEW.booking_id;
    stay_days := check_out_date - check_in_date;
    calculated_total_amount := room_price_per_night * stay_days;
    SELECT COALESCE(SUM(s.price), 0)
    INTO calculated_service_charges
    FROM service s
    WHERE s.booking_id = NEW.booking_id;
    SELECT COALESCE(MAX(p.discount_percentage), 0)
    INTO applicable_discount_rate
    FROM promotion p
    WHERE NEW.issue_date BETWEEN p.start_date AND p.end_date
        AND (p.applicable_room_types IS NULL
            OR p.applicable_room_types = ''
            OR POSITION(room_type_value IN p.applicable_room_types) > 0)
        AND stay_days >= p.minimum_stay_days;
    calculated_discount_amount :=
        (calculated_total_amount + calculated_service_charges) * applicable_discount_rate / 100;
    calculated_tax_amount := calculated_total_amount * 0.10;
    calculated_final_amount :=
        calculated_total_amount + calculated_service_charges + calculated_tax_amount - calculated_discount_amount;
    NEW.total_amount := calculated_total_amount;
    NEW.tax_amount := calculated_tax_amount;
    NEW.service_charges := calculated_service_charges;
    NEW.discount_amount := calculated_discount_amount;
    NEW.final_amount := calculated_final_amount;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

Nội dung chương trình

4.2. Khởi tạo các trigger/function

```
1 CREATE OR REPLACE TRIGGER trigger_calculate_invoice_amounts  
2 BEFORE INSERT OR UPDATE ON Invoice  
3 FOR EACH ROW  
4 EXECUTE FUNCTION calculate_invoice_amounts();  
5  
6 |
```

5. Truy vấn cá nhân

5.1. Truy vấn liên quan đến khách hàng

5.2. Truy vấn liên quan đến quản lý

5.3. Phân tích hiệu năng

Nội dung bài

5.1. Truy vấn liên quan đến khách hàng

1. Tạo view để khách hàng có thể xem thông tin profile của bản thân, với view này quản lí cũng có thể xem được thông tin của các khách hàng (ngoại trừ những thông tin nhạy cảm như mật khẩu)

```
CREATE OR REPLACE FUNCTION get_customer_profile(p_customer_id INTEGER)
RETURNS TABLE(
    customer_id INTEGER,
    full_name VARCHAR(100),
    phone_number CHAR(11),
    email VARCHAR(100),
    nationality VARCHAR(100),
    registration_date DATE,
    membership_level INTEGER,
    account_status_text TEXT,
    membership_level_text TEXT,
    days_since_registration INTEGER
) AS $$ 
BEGIN
    RETURN QUERY
    SELECT cp.customer_id, cp.full_name,
        cp.phone_number, cp.email, cp.nationality,
        cp.registration_date, cp.membership_level, cp.account_status_text,
        cp.membership_level_text, cp.days_since_registration
    FROM customer_profile cp
    WHERE cp.customer_id = p_customer_id;
END; |
$$ LANGUAGE plpgsql SECURITY DEFINER;
```

Nhóm 5: Truy vấn liên quan đến khách hàng

Kết quả:

The screenshot shows a PostgreSQL database interface with a SQL query and its results. The query is:

```
1 SELECT * FROM get_customer_profile(1);
```

The results are displayed in a table:

customer_id	full_name	phone_number	email	nationality	registration_date	membership_level	account_status_text	membership_level_text	days_since_registration
1	Dang Thanh Anh	09127490375	user1@yahoo.com	American	2024-07-19	0	Hoạt động	Thanh viên thường	334

Total rows: 1 Query complete 00:00:00.060

Nội dung bài

5.1. Truy vấn liên quan đến khách hàng

2. Đăng ký tài khoản khách hàng, kiểm tra trùng username/email/CCCD

```
CREATE OR REPLACE FUNCTION register_customer(
    p_full_name VARCHAR,
    p_identification_number CHAR(12),
    p_phone_number CHAR(11),
    p_email VARCHAR,
    p_nationality VARCHAR,
    p_username VARCHAR,
    p_password VARCHAR
) RETURNS TEXT AS $$
DECLARE
    duplicate_count INTEGER;
BEGIN
    SELECT COUNT(*) INTO duplicate_count
    FROM customer
    WHERE username = p_username
    OR email = p_email
    OR identification_number = p_identification_number;
    IF duplicate_count > 0 THEN
        RETURN 'Thông tin đăng ký đã tồn tại!';
    END IF;

    INSERT INTO customer (
        full_name, identification_number, phone_number, email, nationality, username,
        password, registration_date, account_status
    ) VALUES (
        p_full_name, p_identification_number, p_phone_number, p_email, p_nationality,
        p_username, p_password, CURRENT_DATE, 1
    );
    RETURN 'Đăng ký thành công!';
END;
$$ LANGUAGE plpgsql;
```

Nội dung bài

5.1. Truy vấn liên quan đến khách hàng

Kết quả:

```
hotel_management=# SELECT register_customer(
hotel_management(#   'Le Thanh An',
hotel_management(#   '012345678901',
hotel_management(#   '09876543210',
hotel_management(#   'leva@example.com',
hotel_management(#   'Vietnam',
hotel_management(#   'levana01',
hotel_management(#   'matkhau123'
hotel_management(# );
register_customer
-----
Dang ky thanh cong!
(1 row)

[hotel_management=# SELECT register_customer(
[hotel_management(#   'Nguyen Van D',
[hotel_management(#   '222222222222',
[hotel_management(#   '0922222222',
[hotel_management(#   'vand@example.com',
[hotel_management(#   'Vietnam',
[hotel_management(#   'levana01',
[hotel_management(#   'matkhau789'
[hotel_management(# );
register_customer
-----
Thong tin dang ky da ton tai!
(1 row)
```

Nội dung bài

5.1. Truy vấn liên quan đến khách hàng

3. Khách hàng cập nhật thông tin cá nhân

```
CREATE OR REPLACE FUNCTION update_customer_profile(
    p_customer_id INTEGER,
    p_full_name VARCHAR(100) DEFAULT NULL, -- default null vì có thể khách hàng không
    update field này
    p_phone_number CHAR(11) DEFAULT NULL,
    p_email VARCHAR(100) DEFAULT NULL,
    p_nationality VARCHAR(100) DEFAULT NULL
)
RETURNS BOOLEAN AS $$

DECLARE
    record_count INTEGER;
BEGIN
    SELECT COUNT(*) INTO record_count -- đếm số cột bị thay đổi
    FROM customer
    WHERE customer_id = p_customer_id AND account_status >= 0;

    IF record_count = 0 THEN
        RETURN FALSE;
    END IF;

    UPDATE customer
    SET
        full_name = COALESCE(p_full_name, full_name),
        phone_number = COALESCE(p_phone_number, phone_number),
        email = COALESCE(p_email, email),
        nationality = COALESCE(p_nationality, nationality)
    WHERE customer_id = p_customer_id;
    RETURN TRUE;
END;
$$ LANGUAGE plpgsql SECURITY DEFINER;
-- SECURITY DEFINER để đảm bảo người dùng không có truy cập thẳng vào bảng customer
-- mà vẫn có thể thay đổi được thông tin cá nhân của họ
```

Nội dung bài học

5.1. Truy vấn liên quan đến khách hàng

Kết quả:

```
hotel_management=> SELECT update_customer_profile(1, 'Le Van Thanh An 2', '0123456789', 'levanthanhhan@example.com', 'Vietnam');
update_customer_profile
t
(1 row)
```

```
hotel_management=> select * from get_customer_profile(1);
customer_id | full_name      | phone_number | email           | nationality | registration_date | membership_level | account_status_text | membership_level_text | day_s.since_registration
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 | Le Van Thanh An 2 | 0123456789 | levanhanhan@example.com | Vietnam | 2024-07-19 | 0 | Hoat dong | Thanh vien thuong | 334
(1 row)
```

Nội dung bài học

5.1. Truy vấn liên quan đến khách hàng

4. Khách hàng xem hoá đơn của mình

```
CREATE OR REPLACE FUNCTION get_booking_details(p_booking_id INTEGER)
RETURNS TABLE(
    booking_id INTEGER,
    customer_name VARCHAR(100),
    room_number INTEGER,
    room_type VARCHAR(20),
    check_in_date DATE,
    check_out_date DATE,
    nights INTEGER,
    room_price NUMERIC(10,2),
    promotion_name VARCHAR(100),
    total_room_amount NUMERIC(10,2),
    service_charges NUMERIC(10,2),
    tax_amount NUMERIC(10,2),
    discount_amount NUMERIC(10,2),
    final_amount NUMERIC(10,2),
    payment_status INTEGER
) AS $$
BEGIN
    RETURN QUERY
    SELECT
```

Nội dung bài học

5.1. Truy vấn liên quan đến khách hàng

4. Khách hàng xem hoá đơn của mình

```
b.booking_id,  
c.full_name,  
r.room_number,  
r.room_type,  
b.check_in_date,  
b.check_out_date,  
(b.check_out_date - b.check_in_date)::INTEGER AS nights,  
r.price_per_night,  
p.promotion_name,  
COALESCE(i.total_amount, 0)::NUMERIC(10,2),  
COALESCE(i.service_charges, 0)::NUMERIC(10,2),  
COALESCE(i.tax_amount, 0)::NUMERIC(10,2),  
COALESCE(i.discount_amount, 0)::NUMERIC(10,2),  
COALESCE(i.final_amount, 0)::NUMERIC(10,2),  
COALESCE(i.payment_status, 0)::INTEGER  
FROM booking b  
JOIN customer c ON b.customer_id = c.customer_id  
JOIN room r ON b.room_id = r.room_id  
LEFT JOIN promotion p ON b.promotion_id = p.promotion_id  
LEFT JOIN invoice i ON b.booking_id = i.booking_id  
WHERE b.booking_id = p_booking_id;  
END;  
$$ LANGUAGE plpgsql;
```

5.1. Truy vấn liên quan đến khách hàng

Kết quả:

booking_id	customer_name	room_number	room_type	check_in_date	check_out_date	nights	room_price	promotion_name	total_room_amount	service_charges	tax_amount
discount_amount	final_amount	payment_status									
1505	Hoang Gia Cuong	102	Deluxe	2025-06-20	2025-06-22	2	470.00		940.00	72.86	94.00
(1 row)											

Nội dung bài học

5.1. Truy vấn liên quan đến khách hàng

5. Khách hàng huỷ phòng:

```
1 CREATE OR REPLACE FUNCTION cancel_booking_by_customer(
2     p_booking_id INTEGER,
3     p_customer_id INTEGER,
4     p_cancellation_reason VARCHAR(500)
5 )
6 RETURNS TABLE (
7     booking_id INTEGER,
8     message TEXT
9 ) AS $$ 
10 DECLARE
11     v_room_id INTEGER;
12 BEGIN
13     -- Kiểm tra booking tồn tại, đúng khách và đang hoạt động
14     SELECT b.room_id
15     INTO v_room_id
16     FROM booking b
17     WHERE b.booking_id = p_booking_id
18     AND b.customer_id = p_customer_id
19     AND b.status IN (1);
```

Nội dung bài

5.1. Truy vấn liên quan đến khách hàng

5. Khách hàng huỷ phòng:

```
21 IF NOT FOUND THEN
22   RETURN QUERY SELECT NULL, 'Booking khong ton tai hoac khong thuoc ve khach hang nay';
23   RETURN;
24 END IF;
25
26 -- Cập nhật trạng thái booking
27 UPDATE booking b
28 SET status = 0,
29     cancellation_reason = p_cancellation_reason,
30     cancellation_date = CURRENT_DATE
31 WHERE b.booking_id = p_booking_id;
32
33 -- Cập nhật trạng thái phòng về trống
34 UPDATE room r
35 SET status = 0
36 WHERE r.room_id = v_room_id;
37
38 RETURN QUERY SELECT p_booking_id, 'Booking da duoc huy';
39
40 END;
41 $$ LANGUAGE plpgsql;
```

Nhận định và giải quyết vấn đề

5.1. Truy vấn liên quan đến khách hàng

Kết quả:

```
hotel_management=> SELECT * FROM cancel_booking_by_customer(1505, 2, 'Khach thay doi ke hoach');
booking_id |      message
-----+-----
    1505 | Booking da duoc huy
(1 row)

hotel_management=> SELECT * FROM cancel_booking_by_customer(1505, 2, 'Khach thay doi ke hoach');
booking_id |      message
-----+-----
    1505 | Booking da duoc huy
(1 row)

hotel_management=# select room_id, status from room where room_id = 2;
room_id | status
-----+-----
    2 |      0
(1 row)
```

Nhóm 3: Truy vấn liên quan đến quản lý

5.2. Truy vấn liên quan đến quản lí

1. Gợi ý khuyến mãi cho khách hàng:

```
CREATE OR REPLACE FUNCTION suggest_promotions(
    p_room_type VARCHAR,
    p_stay_days INT,
    p_current_date DATE DEFAULT CURRENT_DATE
)
RETURNS TABLE (
    promotion_id INT,
    promotion_name VARCHAR,
    discount_percentage NUMERIC,
    start_date DATE,
    end_date DATE
) AS $$
BEGIN
    RETURN QUERY
    SELECT
        p.promotion_id,
        p.promotion_name,
        p.discount_percentage,
        p.start_date,
        p.end_date
    FROM promotion p
    WHERE p.current_date BETWEEN p.start_date AND p.end_date
        AND (p.applicable_room_types IS NULL OR p.applicable_room_types = '' OR
        p.applicable_room_types = p.room_type)
        AND p.minimum_stay_days <= p_stay_days;
END;
$$ LANGUAGE plpgsql;
```

5.2. Truy vấn liên quan đến quản lý

Kết quả:

```
SELECT * FROM suggest_promotions('Double', 3, '2025-06-20');
```

Data Output Messages Notifications

SQL

	promotion_id integer	promotion_name character varying	discount_percentage numeric	start_date date	end_date date
1	1	Summer Splash	7	2025-06-18	2025-06-23

Nội dung bài 2: Truy vấn

5.2. Truy vấn liên quan đến quản lý

2. Gợi ý phòng trống theo yêu cầu của khách hàng:

```
CREATE OR REPLACE FUNCTION suggest_available_rooms(
    p_room_type VARCHAR,
    p_min_capacity INT,
    p_check_in_date DATE,
    p_check_out_date DATE
)
RETURNS TABLE (
    room_id INT,
    room_number INT,
    capacity INT,
    price_per_night NUMERIC(10,2),
    floor_number INT
) AS $$
BEGIN
    RETURN QUERY
    SELECT
        r.room_id,
        r.room_number,
        r.capacity,
        r.price_per_night,
        r.floor_number
    FROM room r
    WHERE r.room_type = p_room_type
        AND r.capacity >= p_min_capacity
        AND r.status != 2
        AND NOT EXISTS (
            SELECT 1 FROM booking b
            WHERE b.room_id = r.room_id
                AND b.status = 1
                AND (
                    b.check_in_date < p_check_out_date
                    AND b.check_out_date > p_check_in_date
                )
        )
    ORDER BY r.price_per_night;
END;
$$ LANGUAGE plpgsql;
```

5.2. Truy vấn liên quan đến quản lý

Kết quả:

```
SELECT * FROM suggest_available_rooms('Deluxe', 3, '2025-06-10', '2025-06-13');
```

Data Output Messages Notifications

The screenshot shows a database interface with a toolbar at the top labeled "Data Output", "Messages", and "Notifications". Below the toolbar is a table with the following data:

	room_id integer	room_number integer	capacity integer	price_per_night numeric	floor_number integer
1	1	101	3	450.00	2
2	8	108	3	450.00	9
3	15	115	3	450.00	16
4	22	122	3	450.00	3
5	29	129	3	450.00	10
6	36	136	3	450.00	17
7	43	143	3	450.00	4
8	50	150	3	450.00	11
9	57	157	3	450.00	18
10	64	164	3	450.00	5
11	71	171	3	450.00	12
12	78	178	3	450.00	19

Nhóm 3: Tạo view

5.2. Truy vấn liên quan đến quản lí

3. Tạo view đưa ra thông tin 10 ưu đãi mới nhất cho phòng đơn và phòng đôi

```
CREATE OR REPLACE VIEW v_top10_newest_single_double_promotions AS
SELECT *
FROM promotion
WHERE applicable_room_types IN ('Single', 'Double')
OR applicable_room_types IS NULL
ORDER BY start_date DESC
LIMIT 10; |
```

Nhóm 3: Tạo view

5.2. Truy vấn liên quan đến quản lý

Kết quả:

Select * from v_top10_newest_single_double_promotions;

	promotion_id	promotion_name	description	discount_percentage	start_date	end_date	applicable_room_types	minimum_stay_days
1	20	Room Rush	Get 45% off for a limited time!	45	2025-11-05	2025-11-10		
2	17	Midweek Madness	Get 39% off for a limited time!	39	2025-10-15	2025-10-20	Double	
3	16	Limited Luxury	Get 37% off for a limited time!	37	2025-10-08	2025-10-13		
4	13	Sunny Days	Get 31% off for a limited time!	31	2025-09-17	2025-09-22	Double	
5	12	Cozy Nights	Get 29% off for a limited time!	29	2025-09-10	2025-09-15		
6	9	Flash Deal	Get 23% off for a limited time!	23	2025-08-20	2025-08-25	Double	
7	8	Holiday Bonus	Get 21% off for a limited time!	21	2025-08-13	2025-08-18		
8	5	Rainy Season Sale	Get 15% off for a limited time!	15	2025-07-23	2025-07-28	Double	
9	4	VIP Deal	Get 13% off for a limited time!	13	2025-07-16	2025-07-21		
10	1	Summer Splash	Get 7% off for a limited time!	7	2025-06-25	2025-06-30	Double	
	(10 rows)							

Nhập liệu và truy vấn cơ sở dữ liệu

5.2. Truy vấn liên quan đến quản lý

4. Đưa ra các nhà cung cấp(supplier) cho item và số item họ cung cấp theo thứ tự giảm dần:

```
SELECT supplier_info, COUNT(*) AS total_items
FROM inventory
GROUP BY supplier_info
ORDER BY total_items DESC, supplier_info;
```

Nội dung bài 5. Truy vấn liên quan đến quản lý

5.2. Truy vấn liên quan đến quản lý

Kết quả:

```
hotel_management=# SELECT supplier_info, COUNT(*) AS total_items
hotel_management-# FROM inventory
hotel_management-# GROUP BY supplier_info
hotel_management-# ORDER BY total_items DESC, supplier_info;
supplier_info      | total_items
-----+-----
BlueSky Supplies  |      300
Kim Long Ltd.     |      300
Phuc An JSC       |      300
SunHouse Co.      |      300
Thanh Binh Co.    |      300
(5 rows)
```

Nhóm 3: Truy vấn liên quan

5.2. Truy vấn liên quan đến quản lý

5. Đưa ra danh sách vật tư dưới 1 mốc số lượng nhất định

```
SELECT item_id, item_name, quantity_in_stock  
FROM inventory  
WHERE quantity_in_stock < 10  
ORDER BY quantity_in_stock ASC, item_name;
```

Nội dung bài 5.2: Truy vấn liên quan đến quản lý

5.2. Truy vấn liên quan đến quản lý

Kết quả:

```
hotel_management=# SELECT item_id, item_name, quantity_in_stock
hotel_management# FROM inventory
hotel_management# WHERE quantity_in_stock < 10
hotel_management# ORDER BY quantity_in_stock ASC, item_name;
      item_id |      item_name | quantity_in_stock
-----+-----+-----+
      766 | Blanket 766 |          5
     1379 | Curtains 1379 |          5
       40 | Hair Dryer 40 |          5
      625 | Hair Dryer 625 |          5
      85 | Hair Dryer 85 |          5
     264 | Hanger 264 |          5
    1121 | Kettle 1121 |          5
     161 | Kettle 161 |          5
    1258 | Mirror 1258 |          5
     313 | Mirror 313 |          5
     943 | Mirror 943 |          5
    1035 | Pillow 1035 |          5
     435 | Pillow 435 |          5
   1444 | Shampoo 1444 |          5
   1433 | Slippers 1433 |          5
      17 | Towel 17 |          5
     197 | Towel 197 |          5
    1071 | TV 1071 |          5
   1221 | TV 1221 |          5
     246 | TV 246 |          5
    1066 | Blanket 1066 |          6
     192 | Desk Lamp 192 |          6
     657 | Desk Lamp 657 |          6
```

Nhóm 3: Truy vấn liên quan

5.2. Truy vấn liên quan đến quản lý

6. Đưa ra danh sách vật tư mà khách sạn hiện đang có và số lượng theo thứ tự giảm dần

```
SELECT item_id, item_name, quantity_in_stock  
FROM inventory  
ORDER BY quantity_in_stock DESC, item_name;
```

Nhập liệu và truy vấn cơ sở dữ liệu

5.2. Truy vấn liên quan đến quản lý

Kết quả:

```
hotel_management=# SELECT item_id, item_name, quantity_in_stock
hotel_management=# FROM inventory
hotel_management# ORDER BY quantity_in_stock DESC, item_name;
item_id | item_name      | quantity_in_stock
-----+-----+-----+
1422  | Desk Lamp 1422 | 100
402   | Desk Lamp 402  | 100
459   | Hanger 459    | 100
1076  | Kettle 1076   | 100
367   | Minibar 367   | 100
712   | Minibar 712   | 100
538   | Mirror 538    | 100
808   | Mirror 808    | 100
1309  | Shampoo 1309  | 100
1310  | Soap Bar 1310 | 100
243   | Toothbrush Set 243 | 100
723   | Toothbrush Set 723 | 100
797   | Towel 797     | 100
741   | TV 741        | 100
166   | Blanket 166   | 99
691   | Blanket 691   | 99
777   | Desk Lamp 777  | 99
1284  | Hanger 1284   | 99
1374  | Hanger 1374   | 99
487   | Minibar 487   | 99
388   | Mirror 388    | 99
780   | Pillow 780    | 99
```

Nhóm 3: Truy vấn liên quan đến quản lý

5.2. Truy vấn liên quan đến quản lí

7. Quản lý kiểm tra các vật tư có số lượng ít:

```
CREATE OR REPLACE FUNCTION check_low_inventory()
RETURNS TABLE(
item_id INTEGER,
item_name VARCHAR(500),
current_stock INTEGER,
unit_price NUMERIC(10,2),
supplier_info VARCHAR,
location VARCHAR(100),
warning_message TEXT
) AS $$
BEGIN
    RETURN QUERY
SELECT
    i.item_id,
    i.item_name,
    i.quantity_in_stock,
    i.unit_price,
    i.supplier_info,
    i.location_in_storage,
    CONCAT('CẢNH BÁO: ', i.item_name, ' chỉ còn ', i.quantity_in_stock)
        AS warning_message
FROM Inventory i
WHERE i.quantity_in_stock < 20;
END;
$$ LANGUAGE plpgsql;
```

5.2. Truy vấn liên quan đến quản lý

Kết quả:

The screenshot shows a SQL query results table with the following columns: Item_Id, Item_name, current_stock, unit_price, supplier_info, location, and warning_message. The table contains 248 rows of data. A message at the bottom right indicates the query was successfully run and completed in 82 msec.

Item_Id	Item_name	current_stock	unit_price	supplier_info	location	warning_message
17	Towel 17	19	107.35	Bluesky Supplies	Shelf 18 -Zone C	CANH BAO: Towel 17 chi con 19
22	M minibar 22	8	64.32	Bluesky Supplies	Shelf 3 - Zone C	CANH BAO: Minibar 22 chi con 8
25	Hair Dryer 25	18	39.94	Thanh Binh Co.	Shelf 6 - Zone A	CANH BAO: Hair Dryer 25 chi con 18
27	Desk Lamp 27	16	72.89	Bluesky Supplies	Shelf 8 - Zone C	CANH BAO: Desk Lamp 27 chi con 16
29	Curtains 29	9	29.84	Phuc An JSC	Shelf 10 - Zone E	CANH BAO: Curtains 29 chi con 9
31	Blanket 31	10	104.70	Kim Long Ltd.	Shelf 12 - Zone B	CANH BAO: Blanket 31 chi con 10
33	Toothbrush Set 33	14	177.27	SunHouse Co.	Shelf 14 -Zone D	CANH BAO: Toothbrush Set 33 chi con 14
34	Shampoo 34	19	156.51	Phuc An JSC	Shelf 15 -Zone E	CANH BAO: Shampoo 34 chi con 19
37	M minibar 37	18	85.82	Bluesky Supplies	Shelf 18 - Zone C	CANH BAO: Minibar 37 chi con 18
41	Kettle 41	16	146.21	Kim Long Ltd.	Shelf 2 - Zone B	CANH BAO: Kettle 41 chi con 16
46	Blanket 46	12	168.57	Kim Long Ltd.	Shelf 7 - Zone B	CANH BAO: Blanket 46 chi con 12
60	Pillow 60	6	19.45	Thanh Binh Co.	Shelf 1 - Zone A	CANH BAO: Pillow 60 chi con 6
76	Blanket 76	9	79.90	Kim Long Ltd.	Shelf 17 -Zone B	CANH BAO: Blanket 76 chi con 9

Nhóm 3: Truy vấn liên quan đến quản lý

5.2. Truy vấn liên quan đến quản lý

8. Thống kê tổng tiền dịch vụ theo các loại trong mùa du lịch

```
CREATE OR REPLACE FUNCTION services_revenue_by_type_summer_2025()
RETURNS TABLE (
    service_type CHAR(100),
    total_revenue NUMERIC(10,2),
    number_of_services BIGINT
) AS $$ 
BEGIN
    RETURN QUERY
    SELECT
        s.service_type,
        SUM(s.price)::NUMERIC(10,2) AS total_revenue,
        COUNT(*) AS number_of_services
    FROM service s
    JOIN booking b ON s.booking_id = b.booking_id
    WHERE EXTRACT(YEAR FROM b.check_out_date) = 2025
    AND EXTRACT(MONTH FROM b.check_out_date) BETWEEN 6 AND 8
    GROUP BY s.service_type
    ORDER BY total_revenue DESC;
END;
$$ LANGUAGE plpgsql;
SELECT * FROM services_revenue_by_type_summer_2025();
```

Nội dung bài 5.2

5.2. Truy vấn liên quan đến quản lí

Kết quả:

```
SELECT * FROM services_revenue_by_type_summer_2025();
```

Data Output Messages Notifications

	service_type	total_revenue	number_of_services
1	Spa	1000.90	16
2	Laundry	924.85	15
3	Food	902.95	15

(tháng 6,7,8 năm 2025)

Nhóm 3: Truy vấn liên quan đến quản lý

5.2. Truy vấn liên quan đến quản lý

9. Truy vấn tỉ lệ sử dụng dịch vụ của khách hàng

```
SELECT
    TO_CHAR(b.check_out_date, 'YYYY-MM') AS month,
    s.service_type,
    COUNT(*) AS usage_count,
    ROUND(100.0 * COUNT(*)) / SUM(COUNT(*)) OVER (PARTITION BY TO_CHAR(b.check_out_date, 'YYYY-MM')), 2) AS percentage
FROM service s
JOIN booking b ON s.booking_id = b.booking_id
WHERE EXTRACT(YEAR FROM b.check_out_date) = 2025
AND EXTRACT(MONTH FROM b.check_out_date) BETWEEN 5 AND 8
GROUP BY month, s.service_type
ORDER BY month, percentage DESC;
```

Nội dung bài tập 2 - Tạo báo cáo

5.2. Truy vấn liên quan đến quản lý

Kết quả:

	month text	service_type character (100)	usage_count bigint	percentage numeric	
1	2025-05	Food	...	6	37.50
2	2025-05	Laundry	...	5	31.25
3	2025-05	Spa	...	5	31.25
4	2025-06	Food	...	5	33.33
5	2025-06	Laundry	...	5	33.33
6	2025-06	Spa	...	5	33.33
7	2025-07	Spa	...	6	37.50
8	2025-07	Food	...	5	31.25
9	2025-07	Laundry	...	5	31.25
10	2025-08	Food	...	5	33.33
11	2025-08	Laundry	...	5	33.33
12	2025-08	Spa	...	5	33.33

Nhóm 3: Truy vấn liên quan đến quản lý

5.2. Truy vấn liên quan đến quản lý

10. Thống kê top 5 khách hàng có lượng chi tiêu nhiều nhất năm:

```
CREATE OR REPLACE FUNCTION top_spending_customers(year_input INT)
RETURNS TABLE (
    customer_id INT,
    full_name VARCHAR(100),
    total_spent NUMERIC(10,2)
) AS $$ 
BEGIN
    RETURN QUERY
    SELECT
        c.customer_id,
        c.full_name,
        SUM(i.final_amount)::NUMERIC(10,2)
    FROM customer c
    JOIN booking b ON c.customer_id = b.customer_id
    JOIN invoice i ON b.booking_id = i.booking_id
    WHERE EXTRACT(YEAR FROM i.issue_date) = year_input
        AND i.payment_status = 1
    GROUP BY c.customer_id, c.full_name
    ORDER BY SUM(i.final_amount) DESC
    LIMIT 5;
END;
$$ LANGUAGE plpgsql;
```

5.2. Truy vấn liên quan đến quản lý

Kết quả:

```
SELECT * FROM top_spending_customers(2025);
```

Data Output Messages Notifications

	customer_id integer	full_name character varying	total_spent numeric
1	40	Huynh Van Giang	3933.43
2	5	Hoang Van Binh	3930.77
3	75	Nguyen Duc Phuong	3928.36
4	110	Vo Thu Khanh	3899.09
5	19	Pham Thu Phuong	3124.63

Nhóm 3: Truy vấn liên quan đến quản lý

5.2. Truy vấn liên quan đến quản lý

11. Tìm các khách hàng chưa thanh toán hóa đơn:

```
Query  Query History
1 ▼ SELECT
2     i.invoice_id,
3     c.full_name,
4     i.issue_date,
5     i.final_amount,
6     i.payment_method,
7     CURRENT_DATE - i.issue_date AS days_overdue
8 FROM invoice i
9 JOIN booking b ON i.booking_id = b.booking_id
10 JOIN customer_profile c ON b.customer_id = c.customer_id
11 WHERE i.payment_status = 0 AND CURRENT_DATE > i.issue_date
12 ORDER BY i.issue_date ASC;
13
```

5.2. Truy vấn liên quan đến quản lý

Kết quả:

Data Output Messages Notifications							
	invoice_id integer	full_name character varying (100)	issue_date date	final_amount numeric (10,2)	payment_method character (20)	days_overdue integer	
1	5	Pham Thu Phuong	2025-01-13	1145.04	Transfer	163	
2	15	Le Huu Binh	2025-02-04	1348.88	Cash	141	
3	25	Huynh Quoc Hoa	2025-02-22	1576.03	Card	123	
4	35	Bui Thanh Nam	2025-03-16	2034.26	Transfer	101	
5	45	Tran Thu Binh	2025-04-03	620.08	Cash	83	
6	55	Tran Thi Giang	2025-04-25	2137.34	Card	61	
7	65	Bui Minh Nam	2025-05-13	868.25	Transfer	43	
8	75	Tran Thanh Binh	2025-06-04	2201.51	Cash	21	
9	85	Bui Minh Khanh	2025-06-22	694.39	Card	3	

Nội dung bài học

5.2. Truy vấn liên quan đến quản lý

12. Doanh thu theo tháng và theo phòng

```
WITH invoice_with_date AS (
    SELECT
        i.invoice_id,
        i.booking_id,
        i.final_amount,
        EXTRACT(YEAR FROM i.issue_date) AS year,
        EXTRACT(MONTH FROM i.issue_date) AS month
    FROM invoice i
    WHERE i.payment_status = 1
)
SELECT
    iwd.year,
    iwd.month,
    r.room_type,
    SUM(iwd.final_amount) AS total_revenue
FROM invoice_with_date iwd
JOIN booking b ON iwd.booking_id = b.booking_id
JOIN room r ON b.room_id = r.room_id
GROUP BY iwd.year, iwd.month, r.room_type
ORDER BY total_revenue DESC, iwd.year ASC, iwd.month ASC;
```

5.2. Truy vấn liên quan đến quản lý

Kết quả:

	year numeric	month numeric	room_type character varying (20)	total_revenue numeric
1	2025	6	Deluxe	27739.44
2	2026	4	Deluxe	10139.06
3	2031	11	Deluxe	10101.48
4	2025	12	Deluxe	9153.52
5	2029	6	Deluxe	9084.14
6	2030	8	Deluxe	9074.33
7	2025	2	Deluxe	9022.41
8	2028	4	Deluxe	8989.75
9	2033	1	Deluxe	8926.64
10	2031	5	Deluxe	8745.37
11	2030	12	Deluxe	8627.15
12	2028	12	Deluxe	8541.32
13	2029	10	Deluxe	8153.02

Total rows: 491 Query complete 00:00:00.062

Nhóm 3: Truy vấn liên quan đến quản lý

5.2. Truy vấn liên quan đến quản lý

13. Tổng hợp feedback sau khi đặt phòng theo đơn của khách hàng:

```
CREATE OR REPLACE VIEW v_booking_review_summary AS
SELECT
    b.booking_id,
    c.customer_id,
    c.full_name AS customer_name,
    r.room_number,
    b.check_in_date,
    b.check_out_date,
    rv.review_id,
    rv.rating,
    rv.comment,
    rv.review_date,
    rv.response,
    rv.response_date
FROM booking b
JOIN customer c ON b.customer_id = c.customer_id
JOIN room r ON b.room_id = r.room_id
LEFT JOIN review rv ON b.booking_id = rv.booking_id
ORDER BY c.customer_id, b.booking_id;
```

Nhóm 3: Truy vấn liên quan đến quản lý

5.2. Truy vấn liên quan đến quản lý

Kết quả:

```
SELECT * FROM v_booking_review_summary;
```

customer_id	customer_name	room_number	check_in_date	check_out_date	review_id	rating	comment	review_date	r
1500	1 Bui Minh Dung	101	2023-03-20	2023-03-21					
1	2 Vo Huu Hoa	114	2025-01-03	2025-01-05					
2	3 Dang Ngoc Anh	127	2025-01-05	2025-01-08	2	5	Comment sample 3	2025-01-10	
3	4 Bui Thanh Hoa	140	2025-01-07	2025-01-11					
4	5 Huynh Thanh Cuong	153	2025-01-09	2025-01-10					
5	6 Dang Thanh Hoa	166	2025-01-11	2025-01-13					
6	7 Tran Quoc Hoa	179	2025-01-13	2025-01-16					
7	8 Vo Gia Phuong	192	2025-01-15	2025-01-19	7	5	Comment sample 8	2025-01-19	
8	9 Bui Thu Linh	205	2025-01-17	2025-01-18					
9	10 Huynh Ngoc Phuong	218	2025-01-19	2025-01-21	9	5	Comment sample 10	2025-01-23	
10	11 Tran Ngoc Hoa	231	2025-01-21	2025-01-24					

5.3. Phân tích hiệu năng

1. Khi thao tác tìm khách hàng chưa thanh toán hóa đơn

```
EXPLAIN ANALYZE
SELECT
    i.invoice_id,
    c.full_name AS customer_name,
    i.issue_date,
    i.final_amount,
    i.payment_method,
    CURRENT_DATE - i.issue_date AS days_overdue
FROM (
    SELECT invoice_id, booking_id, issue_date, final_amount, payment_method
    FROM invoice
    WHERE payment_status = 0 AND CURRENT_DATE > issue_date
) AS i
JOIN booking b ON i.booking_id = b.booking_id
JOIN customer c ON b.customer_id = c.customer_id
ORDER BY i.issue_date ASC;
```

Planning Time: 20.774 ms
Execution Time: 2.514 ms

5.3. Phân tích hiệu năng

Tối ưu:

```
EXPLAIN ANALYZE
SELECT
    i.invoice_id,
    c.full_name,
    i.issue_date,
    i.final_amount,
    i.payment_method,
    CURRENT_DATE - i.issue_date AS days_overdue
FROM invoice i
JOIN booking b ON i.booking_id = b.booking_id
JOIN customer_profile c ON b.customer_id = c.customer_id
WHERE i.payment_status = 0 AND CURRENT_DATE > i.issue_date
ORDER BY i.issue_date ASC;
```

5.3. Phân tích hiệu năng

Tối ưu:

Việc không sử dụng subquery sẽ nhanh hơn, do trong subquery của câu ban đầu chứa nhiều điều kiện và DBMS sẽ cần phải chạy subquery trước rồi mới thực hiện phép join. Nó làm tăng planning time của câu truy vấn

Cụ thể kết quả trước khi có index

Planning Time: 0.668 ms

Execution Time: 0.433 ms

```
CREATE INDEX IF NOT EXISTS idx_invoice_status_date ON invoice (payment_status, issue_date);
CREATE INDEX idx_invoice_booking_id ON invoice (booking_id);
CREATE INDEX idx_booking_customer_id ON booking (customer_id);
```

sau khi chạy truy vấn, ta thấy truy vấn sử dụng: Bitmap Index Scan on idx_invoice_status_date (nó sẽ truy cập theo từng group, không cần tìm từng dòng), Index Scan using idx_customer_id on customer và Index Cond: (customer_id = b.customer_id). Và câu lệnh cũng cho thấy Postgres đã sử dụng Nested Loop, nó sẽ nhanh với dữ liệu nhỏ và có index

Kết quả sau khi dùng index

Planning Time: 0.410 ms

Execution Time: 0.341 ms

5.3. Phân tích hiệu năng

2. Khi thao tác tính doanh thu:

```
EXPLAIN ANALYZE
WITH invoice_with_date AS (
SELECT
    i.invoice_id,
    i.booking_id,
    i.final_amount,
    EXTRACT(YEAR FROM i.issue_date) AS year,
    EXTRACT(MONTH FROM i.issue_date) AS month
FROM invoice i
WHERE i.payment_status = 1
)
SELECT
    iwd.year,
    iwd.month,
    r.room_type,
    SUM(iwd.final_amount) AS total_revenue
FROM invoice_with_date iwd
JOIN booking b ON iwd.booking_id = b.booking_id
JOIN room r ON b.room_id = r.room_id
GROUP BY iwd.year, iwd.month, r.room_type
ORDER BY total_revenue DESC, iwd.year ASC, iwd.month ASC;
```

5.3. Phân tích hiệu năng

Tối ưu

Ta có thể tách CTE để làm giảm độ phức tạp của truy vấn chính, và cũng có thể sử dụng lại CTE này cho các truy vấn khác (nếu cần). Ngoài ra, ta tránh gọi EXTRACT() nhiều lần, do DBMS không ghi nhớ kết quả EXTRACT trong lệnh GROUP BY. Vậy, cách tốt nhất là tạo CTE là một bảng tạm thời, tồn tại trong phạm vi của một câu truy vấn.

Planning Time: 0.358 ms

Execution Time: 2.895 ms



HUST

THANK YOU !