

**BỘ GIAO DỤC VÀ ĐÀO TẠO**  
**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**Trường Công nghệ thông tin và truyền thông**  
----- o0o -----



**BÁO CÁO THỰC HÀNH**

**ĐỀ TÀI: QUẢN LÝ SÂN CẦU LÔNG**

**Môn Học:** Thực hành Cơ sở Dữ Liệu

**Mã học phần:** IT3290

**Mã lớp:** 147782

**Sinh viên thực hiện:**

Nguyễn Thanh Bình                      MSSV: 20225792

Trương Công Thái Đức                  MSSV: 20225815

Cao Thành Đạt                            MSSV: 20225803

**Giảng viên hướng dẫn:** Nguyễn Thị Oanh



## **LỜI CẢM ƠN**

Lời đầu tiên, chúng em xin được bày tỏ lòng biết ơn chân thành và sâu sắc nhất tới cô giáo NGUYỄN THỊ OANH – người luôn tận tình, nhiệt huyết giúp đỡ, hướng dẫn chúng em trong thời gian học môn Thực hành Cơ sở Dữ Liệu, đã truyền đạt cho chúng em những kiến thức vô cùng hữu ích, giúp chúng em có đủ kiến thức về cơ sở dữ liệu để hoàn thành project này.

## I. LÝ DO CHỌN ĐỀ TÀI

Hệ thống sân cầu lông AB hiện đang có quy trình quản lý một cách rất thủ công, từ việc ghi chú thông tin khách hàng đến lập lịch đặt sân và tạo hóa đơn thanh toán. Sự thủ công này dẫn đến việc mất thời gian và công sức lớn, đồng thời cũng dễ xảy ra sai sót trong quá trình quản lý.

Chính vì vậy, để cải thiện hiệu quả quản lý sân cầu lông AB, cần thiết phải xây dựng một hệ thống quản lý thông tin có cơ sở dữ liệu để tự động hóa quy trình quản lý. Hệ thống quản lý thông tin sẽ giúp tổ chức lưu trữ thông tin khách hàng, nhân viên, sân cầu lông và lịch đặt sân một cách chính xác và tiện lợi. Việc này không chỉ giúp tiết kiệm thời gian và công sức mà còn tạo điều kiện cho việc tăng cường độ chính xác và hiệu quả trong quản lý sân cầu lông AB.

## II. CÁC TÀI NGUYÊN SỬ DỤNG

Ngôn Ngữ	PSQL
----------	------

### III. MÔ TẢ BÀI TOÁN

Hệ thống quản lý sân cầu lông được thiết kế để quản lý thông tin về khách hàng, sân cầu lông, lịch đặt sân và hóa đơn thanh toán. Cụ thể, hệ thống sẽ bao gồm các chức năng sau:

#### Nghiệp vụ

<u>ST</u> <u>T</u>	<u>Actor</u>	<u>Mô tả</u>
1	<u>Khách hàng</u>	<ul style="list-style-type: none"><li>- Có thể đặt một hoặc nhiều sân cho nhiều khung giờ và thanh toán cùng lúc</li><li>- Sẽ được thực hiện các chức năng: tra cứu thông tin liên quan tới sân và kiểm tra xem sân còn trống hay không</li></ul>
2	<u>Nhân Viên</u>	<ul style="list-style-type: none"><li>- Có thể lập và phụ trách các hóa đơn</li><li>- Có quyền quản lý thông tin của sân cầu như kiểm tra các sân còn trống</li></ul>
3	<u>Quản Lý</u>	<ul style="list-style-type: none"><li>- Có những chức năng quyền hạn như nhân viên nhưng sẽ có thể xem thêm doanh thu trong ngày của một ngày bất kì nào đó</li></ul>

•**Quản lý thông tin khách hàng:** Hệ thống quản lý sân cầu lông lưu trữ thông tin về khách hàng. Thông tin khách hàng bao gồm mã khách hàng, tên khách hàng, số điện thoại và địa chỉ. Điều này giúp xác định và liên kết thông tin với mỗi khách hàng, từ đó tiện lợi cho việc xác nhận và liên hệ với họ về các dịch vụ cũng như hóa đơn thanh toán.

•**Quản lý thông tin nhân viên:** Hệ thống quản lý sân cầu lông lưu trữ thông tin về nhân viên. Thông tin nhân viên bao gồm mã nhân viên, tên nhân viên, số điện thoại và địa chỉ, chức vụ. Điều này giúp xác định và liên kết thông tin với mỗi khách hàng, từ đó tiện lợi cho việc xác nhận và liên hệ với họ về các dịch vụ cũng như hóa đơn thanh toán.

•**Quản lý thông tin sân cầu lông:** Dữ liệu về các sân cầu lông bao gồm mã sân, tên sân, vị trí sân và giá tiền. Việc lưu trữ chi tiết này không chỉ giúp quản lý về số lượng sân, mà còn giúp định giá và quản lý chi phí. Thông tin về vị trí sân cũng quan trọng để khách hàng dễ dàng xác định và đến đúng sân mình đã đặt.

•**Thuê sân:** Chức năng này cho phép khách hàng đặt sân theo ngày và giờ mong muốn. Việc quản lý lịch đặt giúp tránh trường hợp trùng lịch và tối ưu hóa sử dụng sân cầu lông. Hơn nữa, việc xác định và thông báo chính xác lịch đặt cũng giúp giảm thiểu việc hủy đặt sân hoặc nhầm lịch đặt tới mức thấp nhất.

•**Tạo hóa đơn thanh toán:** Sau khi khách hàng đặt sân cầu lông và xác nhận lịch đặt, nhân viên sẽ tạo hóa đơn thanh toán. Mỗi lịch thuê sẽ xuất 1 hóa đơn thanh toán và khách thuê sẽ cần phải thanh toán đầy đủ số tiền đã thuê theo hóa đơn.

Ở Tổng thể, hệ thống quản lý sân cầu lông mang lại nhiều lợi ích, bao gồm quản lý hiệu quả thông tin khách hàng, sân cầu lông, lịch đặt sân và hóa đơn thanh toán. Việc tự động hóa quy trình giúp giảm tải công việc cho nhân viên và tạo điều kiện cho việc quản lý và vận hành hệ thống một cách thuận lợi và hiệu quả.

Như vậy, hệ thống quản lý sân cầu lông không chỉ giúp tối ưu hóa quy trình quản lý mà còn cung cấp trải nghiệm thuận lợi và dễ dàng hơn cho cả khách hàng lẫn nhân viên. Đảm bảo tính chính xác và hiệu quả trong việc quản lý sân cầu lông và cung cấp dịch vụ tốt nhất cho người dùng.

#### **IV. Mô hình thực thể liên kết và mô hình quan hệ**

##### **Mối quan hệ**

Một khách hàng có thể đặt nhiều lịch thuê. Một lịch thuê của 1 khách hàng

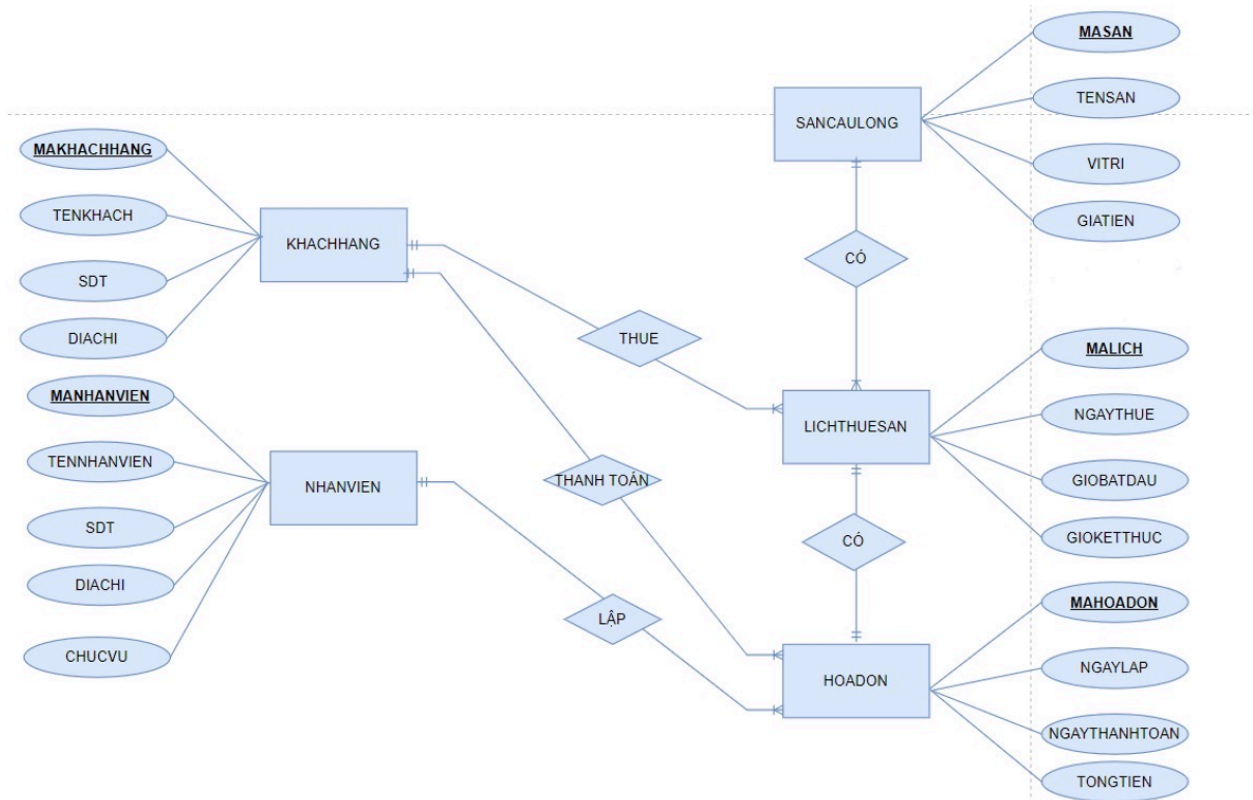
Một sân cầu lông có trong nhiều lịch thuê sân. Một lịch thuê sân chỉ có 1 sân cầu lông

Một hóa đơn của một lịch thuê sân. Một lịch thuê sân xuất ra được 1 hóa đơn duy nhất

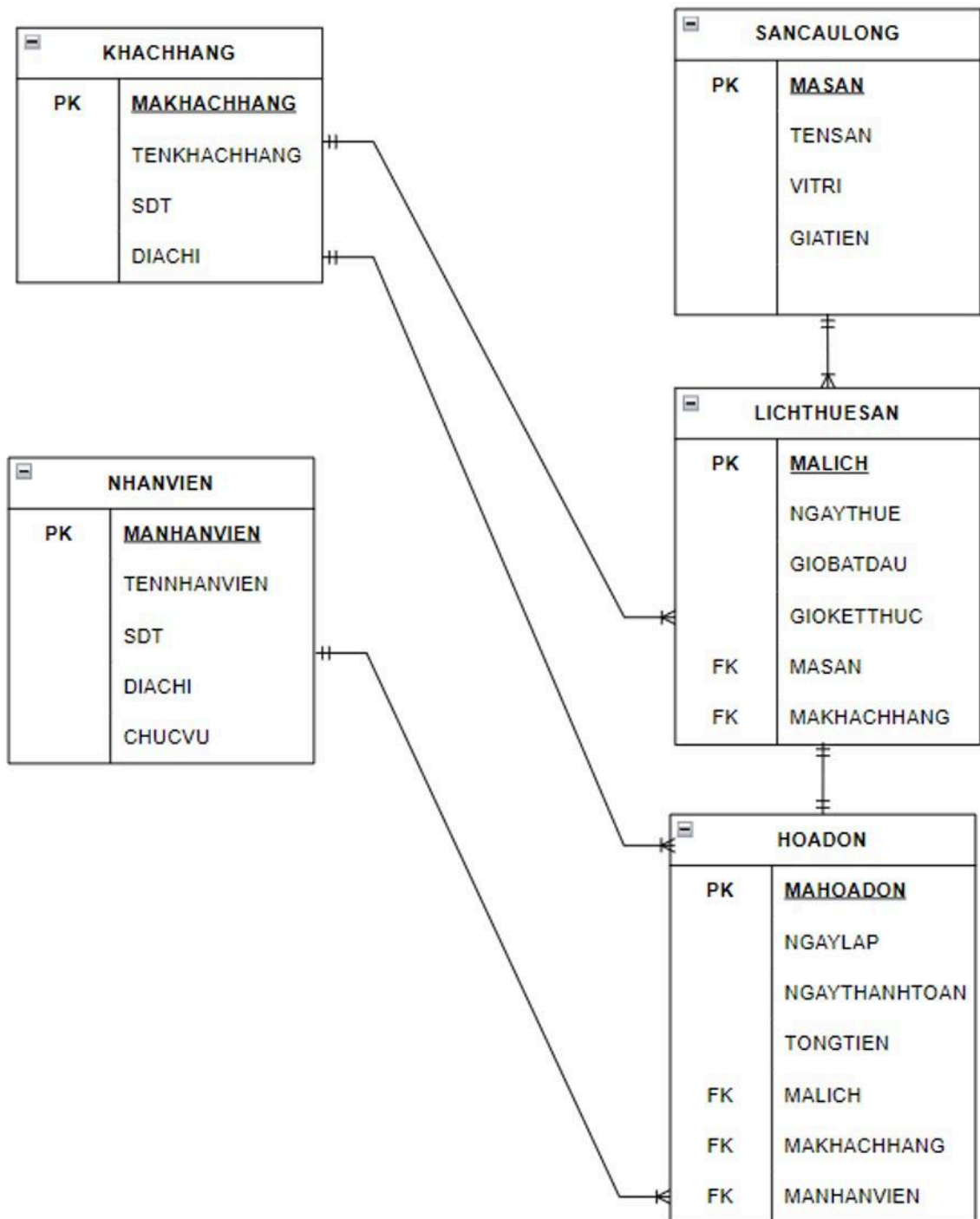
Một khách hàng phải thanh toán nhiều hóa đơn. Một hóa đơn được thanh toán bởi 1 khách hàng

Một nhân viên lập nhiều hóa đơn và 1 hóa đơn chỉ có 1 nhân viên lập

## 1. Mô hình thực thể liên kết



## 2. Mô hình quan hệ





## V. CÂU LỆNH SQL VÀ TRUY VẤN CỦA TỪNG THÀNH VIÊN

### 1. Trương Công Thái Đức - MSSV: 20225815

#### 1. Liệt kê lịch thuê sân với tên khách hàng

```
SELECT KH.TENKHACHHANG, L.*
```

```
FROM LICHTHUESAN L
```

```
INNER JOIN KHACHHANG KH ON L.MAKHACHHANG = KH.MAKHACHHANG;
```

→ Thực hiện hiện phép join 2 bảng cho phép bạn kết hợp thông tin về khách hàng (từ bảng **KHACHHANG**) với thông tin về lịch thuê sân (từ bảng **LICHTHUESAN**).

#### 2. Tìm hóa đơn có tổng tiền lớn hơn 60000:

```
SELECT * FROM HOADON WHERE TONGTIEN > 60000
```

#### 3. Liệt kê các khách hàng có hóa đơn có tổng tiền lớn hơn 70000

```
SELECT
```

```
KHACHHANG.*
```

```

FROM

KHACHHANG

        JOIN    HOADON    ON    KHACHHANG.MAKHACHHANG    =
HOADON.MAKHACHHANG

WHERE

        HOADON.TONGTIEN > 70000;

```

#### **4. Tìm danh sách các khách hàng đã thuê sân trong tháng 6 năm 2024 và tổng số tiền họ đã chi tiêu**

```

WITH tmp AS (

        SELECT

                KH.MAKHACHHANG,

                KH.TENKHACHHANG,

                SUM(HD.TONGTIEN) AS tongchitieu

        FROM

                KHACHHANG KH

        JOIN

                HOADON HD ON KH.MAKHACHHANG = HD.MAKHACHHANG

        JOIN

                CHITIETHOADON CTHD ON HD.MAHOADON = CTHD.MAHOADON

        JOIN

                LICHTHUESAN LTS ON CTHD.MALICH = LTS.MALICH

        WHERE

```

```

EXTRACT(MONTH FROM LTS.NGAYTHUE) = 6

AND EXTRACT(YEAR FROM LTS.NGAYTHUE) = 2024

GROUP BY

KH.MAKHACHHANG, KH.TENKHACHHANG

)

SELECT

MAKHACHHANG,

TENKHACHHANG,

tongchitieu

FROM

tmp

ORDER BY

tongchitieu DESC;

```

Giải thích:

Đoạn mã SQL này nhằm mục đích lấy danh sách các khách hàng và tổng chi tiêu của họ trong tháng 6 năm 2024, sắp xếp từ cao đến thấp dựa trên tổng chi tiêu. Nó sử dụng CTE để trực quan hóa các bước truy vấn và xử lý dữ liệu, giúp cấu trúc câu lệnh trở nên rõ ràng và dễ hiểu hơn.

## 5. Tạo hàm tính doanh thu theo ngày

```
CREATE OR REPLACE FUNCTION tinh_doanh_thu_theo_ngay(ngay DATE)
```

```

RETURNS INT AS $$

DECLARE

    tong_doanh_thu INT;

BEGIN

    SELECT COALESCE(SUM(TONGTIEN), 0) INTO tong_doanh_thu

    FROM HOADON

    WHERE NGAYLAP = ngay;

    RETURN tong_doanh_thu;

END;

$$

LANGUAGE plpgsql;

```

-- Sử dụng hàm để tính doanh thu của ngày 2024-06-01

```
SELECT tinh_doanh_thu_theo_ngay('2024-06-01');
```

Giải thích chức năng:

Hàm **tinh\_doanh\_thu\_theo\_ngay** này có chức năng là tính tổng doanh thu từ các hóa đơn được lập vào một ngày cụ thể được truyền vào qua tham số **ngay**. Nó sử dụng các câu lệnh SQL để truy vấn và tính toán dữ liệu từ bảng **HOADON**, sau đó trả về kết quả tính được dưới dạng một số nguyên (**INT**). Lưu ý rằng trong trường hợp không có hóa đơn nào lập vào ngày cụ thể đó, hàm sẽ trả về **0** thay vì **NULL**, nhờ vào việc sử dụng **COALESCE** để xử lý giá trị **NULL**.

6. Trigger **update\_total\_price** được tạo ra để cập nhật tổng tiền (**TONGTIEN**) của một hóa đơn khi có sự thay đổi trong bảng **CHITIETHOADON**

```
CREATE OR REPLACE FUNCTION update_total_price()

RETURNS TRIGGER AS $$

DECLARE

    v_mahoadon INT;

BEGIN

    -- Lấy mã hóa đơn từ bảng CHITIETHOADON dựa trên mã lịch thuê sản từ NEW

    SELECT MAHOADON INTO v_mahoadon

    FROM CHITIETHOADON

    WHERE MALICH = NEW.MALICH

    LIMIT 1;

    -- Nếu tìm thấy mã hóa đơn

    IF v_mahoadon IS NOT NULL THEN

        -- Cập nhật tổng tiền cho hóa đơn tương ứng

        UPDATE HOADON

        SET TONGTIEN = (

            SELECT SUM(SANCAULONG.GIATIEN)
```

```

FROM CHITIETHOADON

JOIN LICHTHUESAN ON CHITIETHOADON.MALICH = LICHTHUESAN.MALICH

JOIN SANCAULONG ON LICHTHUESAN.MASAN = SANCAULONG.MASAN

WHERE CHITIETHOADON.MAHOADON = v_mahoadon

)

WHERE HOADON.MAHOADON = v_mahoadon;

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

```

```

CREATE TRIGGER update_total_price_trigger

AFTER INSERT OR UPDATE ON CHITIETHOADON

FOR EACH ROW

EXECUTE FUNCTION update_total_price();

```

Giải thích:

Hàm trigger `update_total_price` này được thiết kế để tự động cập nhật trường `TONGTIEN` của các hóa đơn trong bảng `HOADON` dựa trên thông tin từ bảng `CHITIETHOADON`. Khi một bản ghi mới được chèn vào `CHITIETHOADON`, hàm sẽ xác định hóa đơn tương ứng và tính toán lại tổng tiền dựa trên các chi tiết thuê sân, sau đó cập nhật tổng tiền vào hóa đơn đó.

## **7. Tạo trigger để kiểm tra việc chèn ngày thanh toán phải $\geq$ ngày lập hóa đơn**

```
CREATE  
  
OR REPLACE FUNCTION kiem_tra_ngay_thanh_toan () RETURNS TRIGGER AS $$  
  
BEGIN  
  
    IF NEW.NGAYTHANHTOAN < NEW.NGAYLAP THEN  
  
        RAISE EXCEPTION 'Ngày thanh toán phải lớn hơn hoặc bằng ngày lập hóa đơn!';  
  
    END IF;  
  
    RETURN NEW;  
  
END;  
  
$$ LANGUAGE plpgsql;  
  
CREATE TRIGGER tr_kiem_tra_ngay_thanh_toan BEFORE INSERT  
  
OR  
  
UPDATE ON HOADON FOR EACH ROW  
  
EXECUTE FUNCTION kiem_tra_ngay_thanh_toan ();
```

## **8. Tính tổng số lượng hóa đơn mà mỗi khách hàng đã lập**

```
SELECT
```

```

MAKHACHHANG,

COUNT(MAHOADON) AS SO_HOA_DON

FROM

HOADON

GROUP BY

MAKHACHHANG;

```

## 9. Tạo procedure để cập nhật thông tin khách hàng

```

CREATE

OR REPLACE PROCEDURE CapNhatThongTinKhachHang1 (

    IN makh INT,

    IN ten_khachhang VARCHAR(100),

    IN sdt_moi VARCHAR(15),

    IN dia_chi VARCHAR(100)

) AS $$

BEGIN

    UPDATE KHACHHANG

    SET TENKHACHHANG = ten_khachhang, SDT = sdt_moi, DIACHI = dia_chi

```



```
WHERE MAKHACHHANG = makh;
```

```
RAISE NOTICE 'Đã cập nhật thông tin khách hàng thành công!';
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

-- Sử dụng procedure để cập nhật thông tin khách hàng có mã khách hàng là 1

```
CALL CapNhatThongTinKhachHang1 (1, 'Nguyen Van Bao', '098764321', '456 Nguyen  
Hue, Quan 9');
```

**10. Truy vấn con này tìm tất cả các mã nhân viên (**MANHANVIEN**)  
từ bảng **HOADON** mà tổng tiền (**TONGTIEN**) của hóa đơn lớn hơn  
**5000****

```
SELECT TENNHANVIEN  
FROM NHANVIEN  
WHERE MANHANVIEN IN (  
SELECT MANHANVIEN  
FROM HOADON  
WHERE TONGTIEN > 5000  
);
```

## **2. Nguyễn Thanh Bình-MSSV: 20225792**

### **1. Liệt kê lịch thuê sân với tên sân**

```
SELECT S.TENSAN, L.*  
FROM LICHTHUESAN L
```

INNER JOIN SANCAULONG S ON L.MASAN = S.MASAN  
order by L.malich

## **2. Phân loại sân dựa vào giá tiền sân**

```
SELECT TENSAN, GIATIEN,  
  
CASE  
  
    WHEN GIATIEN < 60000 THEN 'VIP_1'  
  
    WHEN GIATIEN >= 60000 AND GIATIEN <= 80000 THEN 'VIP_2'  
  
    ELSE 'VIP_3'  
  
END AS loai_san  
  
FROM SANCAULONG;
```

## **3. Lấy danh sách các nhân viên và số hóa đơn họ đã lập, sắp xếp theo tổng số đơn đã lập giảm dần**

```
SELECT NV.TENNHANVIEN, COUNT(HD.MAHOADON) AS so_lan_phu_trach  
  
FROM NHANVIEN NV  
  
LEFT JOIN HOADON HD ON NV.MANHANVIEN = HD.MANHANVIEN  
  
GROUP BY NV.TENNHANVIEN  
  
ORDER BY COUNT(HD.MAHOADON) DESC;
```

## **4. Tìm các khách hàng đã thanh toán hóa đơn trễ hạn**

WITH LatePayments AS (

SELECT MAHOADON, MAKHACHHANG, NGAYLAP, NGAYTHANHTOAN

FROM HOADON

WHERE NGAYTHANHTOAN > NGAYLAP)

SELECT KH.TENKHACHHANG, LP.MAHOADON, LP.NGAYLAP, LP.NGAYTHANHTOAN

FROM KHACHHANG as KH

JOIN LatePayments as LP ON KH.MAKHACHHANG = LP.MAKHACHHANG;

### **5. Tìm khách hàng chưa từng thuê sân cầu lông**

WITH tmp AS (

SELECT DISTINCT MAKHACHHANG

FROM LICHTHUESAN )

SELECT TENKHACHHANG

FROM KHACHHANG

WHERE MAKHACHHANG NOT IN (SELECT MAKHACHHANG FROM tmp);

### **6. Lấy danh sách các khách hàng đã thuê ít nhất 2 sân cầu lông khác nhau**

SELECT TENKHACHHANG

FROM KHACHHANG

WHERE MAKHACHHANG IN (

SELECT MAKHACHHANG

FROM LICHTHUESAN

GROUP BY MAKHACHHANG

HAVING COUNT(DISTINCT MASAN) >= 2 );

## 7. Lấy thông tin của các sân cầu lông đã được thuê vào một ngày cụ thể

```
SELECT DISTINCT TENSAN, VITRI  
  
FROM SANCAULONG  
  
WHERE MASAN IN (  
  
    SELECT MASAN  
  
    FROM LICHTHUESAN  
  
    WHERE NGAYTHUE = '2024-06-01' );
```

## 8. Tạo hàm tính số lần thuê sân của một khách hàng

```
CREATE OR REPLACE FUNCTION tinh_so_lan_thue_san(makh INT) -- đếm số lượng hàng  
RETURNS INT AS  
$$  
DECLARE  
    so_lan_thue_san INT;  
BEGIN  
    SELECT COALESCE(COUNT(*), 0) INTO so_lan_thue_san  
    FROM LICHTHUESAN  
    WHERE MAKHACHHANG = makh;  
  
    RETURN so_lan_thue_san;  
END;  
$$  
LANGUAGE plpgsql;
```

Giải thích: Hàm `tinh_so_lan_thue_san` trong PostgreSQL được sử dụng để tính số lần một khách hàng đã thuê sân từ bảng `LICHTHUESAN`. Hàm này nhận vào mã khách hàng (`makh`) và đếm số lần xuất hiện của `makh` trong bảng `LICHTHUESAN`, gán vào biến `so_lan_thue_san`. Hàm này sử dụng `COALESCE` để đảm bảo rằng nếu không có bản ghi nào thỏa mãn điều kiện, nó sẽ trả về 0.

**9. --Tạo procedure để hiển thị thông tin khách hàng khi nhập mã khách hàng**

```
CREATE OR REPLACE PROCEDURE HienThiThongTinKhachHang (IN makh INT)
```

```
AS
```

```
$$
```

```
DECLARE
```

```
    khách_hang_info RECORD;
```

```
BEGIN
```

```
    SELECT * INTO khách_hang_info
```

```
    FROM KHACHHANG
```

```
    WHERE MAKHACHHANG = makh;
```

```
    IF FOUND THEN
```

```
        RAISE NOTICE 'Thong tin khách hàng - Ma: %, Ten: %, So dien thoai: %,Dia chi: %',
```

```
        khách_hang_info.MAKHACHHANG,khách_hang_info.TENKHACHHANG,khách_hang_info.SDT, khách_hang_info.DIACHI;
```

```
    ELSE
```

```
        RAISE EXCEPTION 'Khong tìm thấy thông tin cho mã khách hàng %', makh;
```

```
    END IF;
```

```
END;
```

```
$$
```

```
LANGUAGE plpgsql;
```

Giải thích: Đoạn mã PL/pgSQL trên định nghĩa một procedure có tên HienThiThongTinKhachHang, được thiết kế để hiển thị thông tin của một khách hàng dựa trên mã khách hàng (makh) được cung cấp.

Khi thực thi, procedure này sẽ thực hiện các bước sau:

- Tìm kiếm và lấy thông tin chi tiết của khách hàng từ bảng KHACHHANG dựa trên makh.
- Nếu tìm thấy khách hàng, thông tin chi tiết sẽ được in ra dưới dạng thông báo sử dụng lệnh RAISE NOTICE.
- Nếu không tìm thấy khách hàng với mã đã cho, procedure sẽ đưa ra một exception thông báo lỗi.

## 10. Tạo trigger kiểm tra lịch đặt sân có bị trùng nhau

```
CREATE OR REPLACE FUNCTION kiem_tra_trung_lich_thue()
RETURNS TRIGGER AS $$
BEGIN
    -- Kiểm tra sự trùng lặp thời gian thuê
    IF EXISTS (
        SELECT 1
        FROM lichthuesan
        WHERE masan = NEW.masan
        AND ngaythue = NEW.ngaythue
        AND (
            (NEW.giobatdau < gioketthuc AND NEW.gioketthuc > giobatdau)
            OR (NEW.giobatdau >= giobatdau AND NEW.giobatdau < gioketthuc)
            OR (NEW.gioketthuc > giobatdau AND NEW.gioketthuc <= gioketthuc)
        )
    ) THEN
        RAISE EXCEPTION 'Sân % đã được đặt trong khoảng thời gian này. Vui lòng chọn thời gian khác.', NEW.MASAN;
    END IF;

    RETURN NEW;
END;
```

```
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER kiem_tra_trung_lich_thue_trigger  
BEFORE INSERT ON LICHTHUESAN  
FOR EACH ROW  
EXECUTE FUNCTION kiem_tra_trung_lich_thue();
```

Giải thích : Đoạn mã PL/pgSQL và trigger trên được sử dụng để kiểm tra và ngăn chặn sự trùng lặp thời gian thuê sân trong bảng LICHTHUESAN khi có thêm dữ liệu mới. Function kiem\_tra\_trung\_lich\_thue sử dụng câu lệnh IF EXISTS để kiểm tra xem có bất kỳ lịch thuê sân nào trùng lặp thời gian với dữ liệu mới (NEW) không. Điều kiện kiểm tra bao gồm so sánh thời gian bắt đầu và kết thúc của lịch thuê mới với các lịch thuê đã có trong cơ sở dữ liệu. Nếu phát hiện trùng lặp, trigger sẽ kích hoạt và đưa ra một exception thông báo cho người dùng biết sân đã được đặt trong khoảng thời gian đó và yêu cầu chọn thời gian khác. Điều này giúp đảm bảo tính toàn vẹn dữ liệu và tránh tình trạng trùng lặp trong quá trình quản lý lịch thuê sân.

### **3.Cao Thành Đạt - MSSV:20225803**

#### **1.Lấy ra lịch thuê sân vào ngày 6/3/2024**

```
SELECT * FROM LICHTHUESAN WHERE NGAYTHUE=' 2024-03-06';
```

#### **2.Tính tổng số lượng hóa đơn mà mỗi khách hàng đã lập**

```
SELECT  
MAKHACHHANG,COUNT(MAHOADON) AS SO_HOA_DON  
FROM  
HOADON  
GROUP BY  
MAKHACHHANG;
```

#### **3.Tính số lượng các sân cầu lông đã được thuê ít nhất 3 lần**

```
SELECT COUNT(*)  
FROM  
(SELECT MASAN  
FROM LICHTHUESAN  
GROUP BY  
MASAN
```

```
HAVING  
COUNT(DISTINCT MALICH) >= 3  
) AS subquery;
```

#### **4.Tìm sân được sử dụng nhiều nhất**

```
SELECT S.*,COUNT(MALICH) AS SO_LAN_SU_DUNG  
FROM SANCAULONG S LEFT JOIN LICHTHUESAN LTS USING(MASAN)  
GROUP BY S.MASAN  
ORDER BY SO_LAN_SU_DUNG DESC  
LIMIT 1;
```

#### **5.Tính trung bình giá sân được sử dụng trong tháng 6/2024**

```
SELECT  
AVG(S.GIATIEN) AS GiaSanTrungBinhThang6  
FROM  
LICHTHUESAN LTS  
JOIN SANCAULONG S USING(MASAN)  
WHERE  
LTS.NGAYTHUE >= DATE '2024-06-01'  
AND LTS.NGAYTHUE <= DATE '2024-06-30';
```

**Giải thích:** Câu truy vấn này được sử dụng nhằm biết được nhu cầu sử dụng sân của khách hàng trong một tháng( ví dụ như tháng 6) từ đó tạo ra các sự kiện giảm giá hoặc khuyến mãi làm tăng doanh thu.Ta sử dụng hàm AVG() để tính trung bình giá tiền và chỉ lấy các bản ghi trong tháng 6/2024 (điều kiện ngày thuê phải từ 1/6/2024 -> 30/6/2024).

**6.Tạo view cung cấp thông tin chi tiết về từng lần đặt sân của khách hàng, bao gồm tên khách hàng, ngày đặt, giờ bắt đầu, giờ kết thúc và tên sân.**

```
CREATE VIEW ChiTietDatSanKhachHang AS  
SELECT  
KH.TENKHACHHANG AS TenKhachHang,  
LTS.NGAYTHUE AS NgayThue,
```



```

LTS.GIOBATDAU AS GioBatDau,
LTS.GIOKETTHUC AS GioKetThuc,
SC.TENSAN AS TenSan
FROM
    LICHTHUESAN LTS
JOIN
    KHACHHANG KH ON LTS.MAKHACHHANG = KH.MAKHACHHANG
JOIN
    SANCAULONG SC ON LTS.MASAN = SC.MASAN;

```

**7.Tạo view cung cấp lịch sử chi tiết của tất cả các lần đặt sân, bao gồm thông tin khách hàng, thông tin sân và thời gian đặt.**

```

CREATE VIEW LichSuDatSanChiTiet AS
SELECT
    LTS.MALICH AS MaLich,
    KH.TENKHACHHANG AS TenKhachHang,
    SC.TENSAN AS TenSan,
    LTS.NGAYTHUE AS NgayThue,
    LTS.GIOBATDAU AS GioBatDau,
    LTS.GIOKETTHUC AS GioKetThuc
FROM
    LICHTHUESAN LTS
JOIN
    KHACHHANG KH ON LTS.MAKHACHHANG = KH.MAKHACHHANG
JOIN
    SANCAULONG SC ON LTS.MASAN = SC.MASAN;

```

**Giải thích:** View LichSuDatSanChiTiet nhằm cung cấp lịch sử chi tiết của tất cả các lần đặt sân nhằm quản lý và theo dõi các hoạt động đặt sân, qua đó đảm bảo sự chính xác và hiệu quả trong quản lý sân. Trong view, các cột được chọn từ LichThueSan (MaLich, NgayThue, GioBatDau, GioKetThuc), KháchHang (TenKhachHang) và SanCauLong (TenSan).

**8.Tìm mã sân đã được đặt bởi khách hàng có mã là 1.**

```

SELECT TENSAN, VITRI, GIATIENTEN

```

```

FROM SANCAULONG
WHERE MASAN IN (
    SELECT MASAN
    FROM LICHTHUESAN
    WHERE MAKHACHHANG = 1
);

```

### **9.Cập nhật giá sân**

```

CREATE OR REPLACE PROCEDURE update_gia_san(
IN ma_san INT,
IN ten_san VARCHAR(100),
IN gia_moi INT)
LANGUAGE plpgsql
AS $$
BEGIN
    UPDATE SANCAULONG
    SET TENSAN= ten_san,GIATIEN=gia_moi
    WHERE MASAN = ma_san;

    RAISE NOTICE' Cập nhật thành công!';
END;
$$;

```

#### **Giải thích:**

Khi thực thi, procedure này sẽ thực hiện các bước sau:

- Tìm kiếm và cập nhật giá sân từ bảng SANCAULONG dựa trên ma\_san..
- Khi cập nhật thành công , ta sẽ ra in ra lệnh RAISE NOTICE.

### **10.Xóa lịch thuê sân khi khách muốn hủy**

```

CREATE OR REPLACE PROCEDURE xoa_lich(
IN ma_lich INT
)LANGUAGE plpgsql
AS $$
BEGIN
    DELETE FROM LICHTHUESAN

```

```

WHERE MALICH=ma_lich;

RAISE NOTICE 'Xử lý thành công!';
END;
$$;

```

## VI. Tối ưu câu truy vấn bằng index

Tạo index cho thuộc tính makhachhang trong bảng lichthuesan

```
CREATE INDEX idx_lichthuesan_makhachhang ON LICHTHUESAN (MAKHACHHANG);
```

### Trước khi sử dụng index

	QUERY PLAN	
	text	
1	Seq Scan on lichthuesan (cost=0.00..23.60 rows=1360 width=32) (actual time=1.087..1.088 rows=18 loops...	
2	Planning Time: 1.770 ms	
3	Execution Time: 1.096 ms	

```

Successfully run. Total query runtime: 70 msec.
3 rows affected.

```

### Sau khi sử dụng index

	<b>QUERY PLAN</b> text 
1	Seq Scan on lichthuesan (cost=0.00..1.18 rows=18 width=32) (actual time=0.019..0.021 rows=18 loops...
2	Planning Time: 0.919 ms
3	Execution Time: 0.043 ms

Successfully run. Total query runtime: 60 msec.  
3 rows affected.

## VII. Những khó khăn và giải pháp

### Khó khăn: Thiết kế cơ sở dữ liệu hiệu quả

#### Giải pháp:

- **Phân tích ERD:** Sử dụng sơ đồ ERD (Entity-Relationship Diagram) để xác định các thực thể (entities) và mối quan hệ (relationships) giữa chúng. Các thực thể có thể bao gồm Người dùng, Sân, Đặt sân, Thanh toán, v.v.
- **Thiết kế logic:** Thiết kế bảng và các khóa chính/phụ để đảm bảo tính nhất quán và hiệu quả trong việc truy xuất dữ liệu.
- **Tối ưu:** Để tránh việc mỗi lịch thuê sân tương ứng với một dòng trong bảng hóa đơn thì nhóm đã sử dụng thêm một bảng chitiethoadon để có thể gộp những đơn hàng mà khách hàng đặt nhiều lịch thuê sân và trả tiền trong cùng một lần

## **VIII. Kết luận**

### **1 Về dự án**

Mặc dù khối lượng công việc là rất lớn, nhưng bọn em đã cố gắng hoàn thiện và có thể tự tin rằng dự án này có thể đi vào hoạt động một cách trơn tru và hiệu quả.

### **2 Những gì đã học được qua dự án**

Qua dự án này và quá trình học môn “Thực hành Cơ sở dữ liệu”, bọn em đã có thêm rất nhiều kiến thức bổ ích về chuyên ngành IT như: Hệ quản trị CSDL Postgres, Đồng thời, bọn em còn học được cách làm việc nhóm, các phát triển dự án và làm việc theo tiến độ một cách khoa học. Đó sẽ là những hành trang tuyệt vời cho chúng em tự tin và hoàn thành tốt các môn học sắp tới.

---

HẾT

---