



BÁO CÁO ĐỒ ÁN QUẢN LÝ THÔNG TIN

TÊN ĐỀ TÀI : TÌM HIỂU HỆ QUẢN TRỊ CSDL POSTGRESQL

GVHD : TS - NGUYỄN TUẤN ANH
CN - TRẦN QUỐC KHÁNH

THÀNH VIÊN

20521634 - Phạm Nguyên Nam

21522194 - Nguyễn Phúc Khang

NỘI DUNG

I.TÌM HIỂU TỔNG QUAN

II.ĐẶC ĐIỂM NỔI BẬT

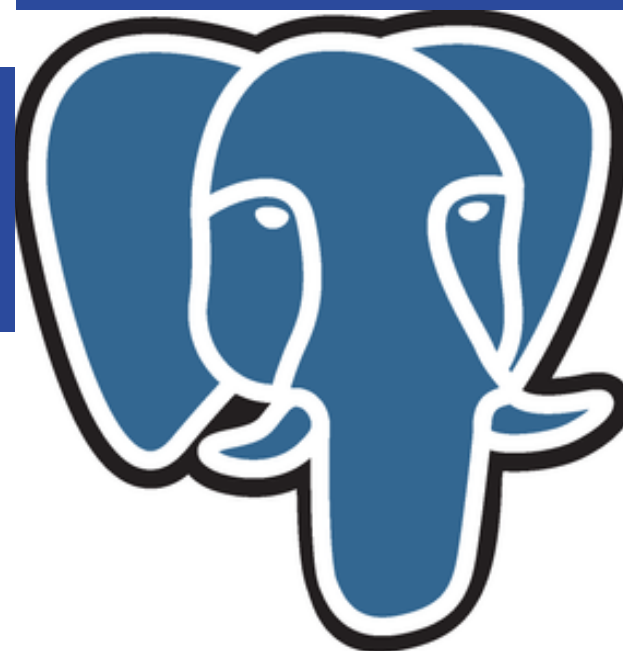
III.CÁC CÂU LỆNH TRONG SQL

IV.Demo CSDL QUẢN LÝ RẠP CHIẾU PHIM

V.ĐÁNH GIÁ VÀ KẾT LUẬN

I

TÌM HIỂU TỔNG QUAN



PostgreSQL
the world's most advanced open source database

I. TÌM HIỂU TỔNG QUAN

1. KHÁI NIỆM

- PostgreSQL là một hệ quản trị cơ sở dữ liệu quan hệ và đối tượng, là hệ thống cơ sở dữ liệu mã nguồn mở miễn phí tiên tiến và được sử dụng phổ biến rộng rãi hiện nay.



- Được phát triển dựa trên POSTGRES 4.2 tại đại học California vào những năm 1989. Ban đầu chỉ được thiết kế chạy trên hệ điều hành UNIX, về sau đã được tương thích trên nhiều hệ điều hành khác như Mac OS, WINDOW, ...



II

ĐẶC ĐIỂM NỔI BẬT



II. ĐẶC ĐIỂM NỔI BẬT

- Mã nguồn mở, miễn phí và phổ biến.
- Hỗ trợ đa nền tảng
- Postgre SQL có khả năng xử lý và quản lý dữ liệu lớn với hiệu suất cao.
- Nó hỗ trợ các tính năng ACID
- Postgre SQL hỗ trợ các tính năng mở rộng.
- Cộng đồng mạnh mẽ và tích hợp nhiều công cụ và thư viện hỗ trợ phát triển ứng dụng.
- Tính ổn định và độ tin cậy cao.

III

CÁC CÂU LỆNH TRONG SQL

```
44 GO
45 SELECT p.Name AS ProductName,
46 NonDiscountSales = (OrderQty *
47 Discounts = ((OrderQty * UnitPr
48 FROM Production.Product AS p
49 INNER JOIN Sales.SalesOrderDetail
50 ON p.ProductID = sod.ProductID
51 ORDER BY ProductName DESC;
52 GO
```


VIEW TABLE

1. ĐỊNH NGHĨA :

- Là một bảng ảo trong cơ sở dữ liệu có nội dung được định nghĩa thông qua câu lệnh SQL.
- Trong PostgreSQL có 4 loại view chính

- **View thông thường**: cho phép chúng ta lưu trữ các câu truy vấn SQL phức tạp nhưng không cho phép bạn cập nhật dữ liệu trong bảng chính
- **View vật liệu** : sử dụng để lưu kết quả tính toán của một truy vấn, giống như bảng thực tế
- **Updatable view** : cho phép người dùng cập nhật trực tiếp trên bảng view
- **View đệ quy** : cho phép thực hiện các việc lên kết đệ quy với chính chúng

VIEW TABLE

- View thông thường

```
61 CREATE VIEW KH_LC(KHACHHANG, LOAICAY)
62 AS
63 SELECT MAKH, TENKH, MALC, TENLC
64 FROM KHACHHANG, LOAICAY;
```

Data Output Explain Messages Notifications

CREATE VIEW

Query returned successfully in 305 msec.

- View vật liệu

```
CREATE MATERIALIZED VIEW DOANH_THU_2022 AS
SELECT RAP.MaRap, RAP.TenRap, EXTRACT(MONTH FROM LC.NgayChieu) AS "Thang", SUM(LC.TongTien) AS "Doanh Thu"
FROM LichChieu AS LC
LEFT JOIN RAP ON RAP.MaRap = LC.MaRap
WHERE EXTRACT(YEAR FROM LC.NgayChieu) = 2022
GROUP BY RAP.MaRap, RAP.TenRap, EXTRACT(MONTH FROM LC.NgayChieu);
```

```
SELECT * FROM DOANH_THU_2022
```

Output Messages Notifications



marap	tenrap	Thang	Doanh Thu
character varying	character varying	numeric	money
BHD02	BHD Star Vũng Tàu	1	\$0.00
BHD02	BHD Star Vũng Tàu	2	\$0.00
BHD02	BHD Star Vũng Tàu	3	\$0.00

- Updatable view

```
CREATE VIEW CHECK_KH
AS
SELECT MAKH, TENKH, DIACHI
FROM KHACHHANG
WHERE DIACHI IS NOT NULL
WITH CHECK OPTION
```

- View đệ quy

```
CREATE VIEW view_name
AS
    WITH RECURSIVE cte_name(columns) AS
    (
        --LOGIC
    )
    SELECT COLUMNS FROM cte_name
```

INDEX

1. ĐỊNH NGHĨA :

- Là một cấu trúc dữ liệu được dùng để định vị và truy vấn nhanh nhất dữ liệu trong các bảng database, tăng thời gian hiệu suất và truy vấn dữ liệu
- Có 6 loại INDEX chính :
 - B-tree Index
 - Hash Index
 - GiST Index
 - GIN Index
 - SP-GiST Index
 - BRIN Index

• Giao diện

The screenshot shows a database management interface for creating an index. The 'Definition' tab is active. The 'Access Method' is set to 'heap'. A list of available index methods is displayed: heap, btree, hash, gist, gin, spgist, and brin. The 'Columns' section is currently empty, and a red error message at the bottom states: "You must specify at least one column." The interface includes buttons for 'Close', 'Reset', and 'Save'.

Access Method	Fill factor	Unique?	Clustered?	Concurrent build?	Constraint
heap	heap				
btree					
hash					
gist					
gin					
spgist					
brin					

Columns

Column	Operator class	Sort order	ROLES	Collation
--------	----------------	------------	-------	-----------

Include columns: Select the column(s)

! You must specify at least one column.

Close Reset Save

INDEX

1. ĐỊNH NGHĨA :

- Là một cấu trúc dữ liệu được dùng để định vị và truy vấn nhanh nhất dữ liệu trong các bảng database, tăng thời gian hiệu suất và truy vấn dữ liệu
- Có 6 loại INDEX chính :
 - B-tree Index
 - Hash Index
 - GiST Index
 - GIN Index
 - SP-GiST Index
 - BRIN Index

• Câu lệnh

B-Tree : `CREATE INDEX KH_MAKH ON KHACHHANG USING BTREE(TEN);`

Hash : `CREATE INDEX KH_MAKH ON KHACHHANG USING HASH(TEN);`

GiST : `CREATE INDEX KH_MAKH ON KHACHHANG USING GiST(TEN TOAN_TU_HO_TRO);`

GIN : `CREATE INDEX KH_MAKH ON KHACHHANG USING GIN(TEN HAM_HO_TRO);`

SP-GiST `CREATE INDEX KH_MAKH ON KHACHHANG USING SPGiST(TEN HAM_HO_TRO);`

BRIN : `CREATE INDEX KH_MAKH ON KHACHHANG USING BRIN(TEN gist_trgm_ops);`

TRIGGER

• CÚ PHÁP

```
CREATE [ CONSTRAINT ] TRIGGER name { BEFORE | AFTER | INSTEAD OF } { event [ OR ... ] }  
ON table_name  
[ FROM referenced_table_name ]  
[ NOT DEFERRABLE | [ DEFERRABLE ] { INITIALLY IMMEDIATE | INITIALLY DEFERRED } ]  
[ FOR [ EACH ] { ROW | STATEMENT } ]  
[ WHEN ( condition ) ]  
EXECUTE PROCEDURE function_name ( arguments )
```

• XÓA TRIGGER

```
DROP TRIGGER my_trigger ON my_table USING FUNCTION my_function;
```

• CẬP NHẬP SỐ VÉ ĐÃ BÁN VÀ TỔNG SỐ TIỀN

```
CREATE OR REPLACE FUNCTION CheckDateLichChieu() RETURNS TRIGGER AS $$  
DECLARE  
    idDinhDang VARCHAR(50); ThoiGianChieu DATE; NgayKhoiChieu DATE; NgayKetThuc DATE;  
BEGIN  
    SELECT MaPhim, DATE(NgayChieu) INTO idDinhDang, ThoiGianChieu FROM NEW;  
  
    SELECT P.NgayKhoiChieu, P.NgayKetThuc INTO NgayKhoiChieu, NgayKetThuc  
    FROM Phim P, LichChieu  
    WHERE idDinhDang = LichChieu.MaPhim AND LichChieu.MaPhim = P.MaPhim;  
  
    IF (ThoiGianChieu > NgayKetThuc OR ThoiGianChieu < NgayKhoiChieu) THEN  
        RAISE EXCEPTION 'Lịch Chieu lớn hơn hoặc bằng Ngày Khởi Chieu và nhỏ hơn hoặc bằng Ngày Kết Thúc';  
    END IF;  
    RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;  
CREATE TRIGGER UTG_INSERT_CheckDateLichChieu  
AFTER INSERT OR UPDATE ON LichChieu  
FOR EACH ROW  
EXECUTE FUNCTION CheckDateLichChieu();
```

• XÓA TRIGGER

```
DROP TRIGGER IF EXISTS UTG_INSERT_CheckDateLichChieu ON LichChieu;
```


FUNCTION

- CÚ PHÁP

```
CREATE FUNCTION function_name (var1 type, var2 type)
  RETURNS type AS
  BEGIN
    --LOGIC
  END;
LANGUAGE language_name;
```

- GỌI HÀM

```
SELECT function_name();
```

- XÓA HÀM

```
DROP FUNCTION function_name();
```

- ĐẾM SỐ LƯỢNG NHÂN VIÊN

```
1 CREATE OR REPLACE FUNCTION getStaffCount()
2 RETURNS integer
3 AS
4 $$
5 BEGIN
6     RETURN (SELECT count(*) FROM staff);
7 END
8 $$
9 LANGUAGE plpgsql;
10
11 SELECT getStaffCount();
```

- KẾT QUẢ TRẢ VỀ

	Data Output	Explain	Messages	Notifications
	getstaffcount integer			
1	12			

STORED PROCEDURE

- CÚ PHÁP

```
create [or replace] procedure procedure_name(parameter_list)
language plpgsql
as $$
    declare
        -- định nghĩa biến
    begin
        -- hàm xử lý
    end; $$
```

- GỌI STORED PROCEDURE

```
call procedure_name(argument_list);
```

- XÓA STORED PROCEDURE

```
DROP PROCEDURE procedure_name();
```

- STORED PROCEDURE NHẬP THÊM TÊN PHIM

```
CREATE OR REPLACE PROCEDURE AddPhim
(_MaPhim VARCHAR(10), _TenPhim VARCHAR(50), _MaHangSX VARCHAR(10), _DaoDien VARCHAR(20), _MaTheLoai VARCHAR(10), _MaNuocSX VARCHAR(10), _NgayKhoiChieu VARCHAR(20), _NgayKetThuc VARCHAR(20), _TongThu VARCHAR(20))
LANGUAGE plpgsql
AS $$
BEGIN
    IF NOT EXISTS (SELECT * FROM NuocSX WHERE MaNuocSX = _MaNuocSX)
    THEN
        RAISE EXCEPTION 'Mã nước sản xuất không tồn tại';
    END IF;
    IF NOT EXISTS (SELECT * FROM HangSX WHERE MaHangSX = _MaHangSX)
    THEN
        RAISE EXCEPTION 'Mã hãng sản xuất không tồn tại';
    END IF;
    IF NOT EXISTS (SELECT * FROM TheLoai WHERE MaTheLoai = _MaTheLoai)
    THEN
        RAISE EXCEPTION 'Mã thể loại không tồn tại';
    END IF;
    IF EXISTS (SELECT * FROM Phim WHERE MaPhim = _MaPhim)
    THEN
        RAISE EXCEPTION 'Phim đã tồn tại';
    END IF;
    INSERT INTO Phim (MaPhim, TenPhim, MaHangSX, DaoDien, MaTheLoai, MaNuocSX, NgayKhoiChieu, NgayKetThuc, TongThu)
    VALUES (_MaPhim, _TenPhim, _MaHangSX, _DaoDien, _MaTheLoai, _MaNuocSX, _NgayKhoiChieu, _NgayKetThuc, _TongThu);
    RAISE NOTICE 'Dữ liệu đã được thêm vào thành công';
END;
```

- GỌI STORED PROCEDURE

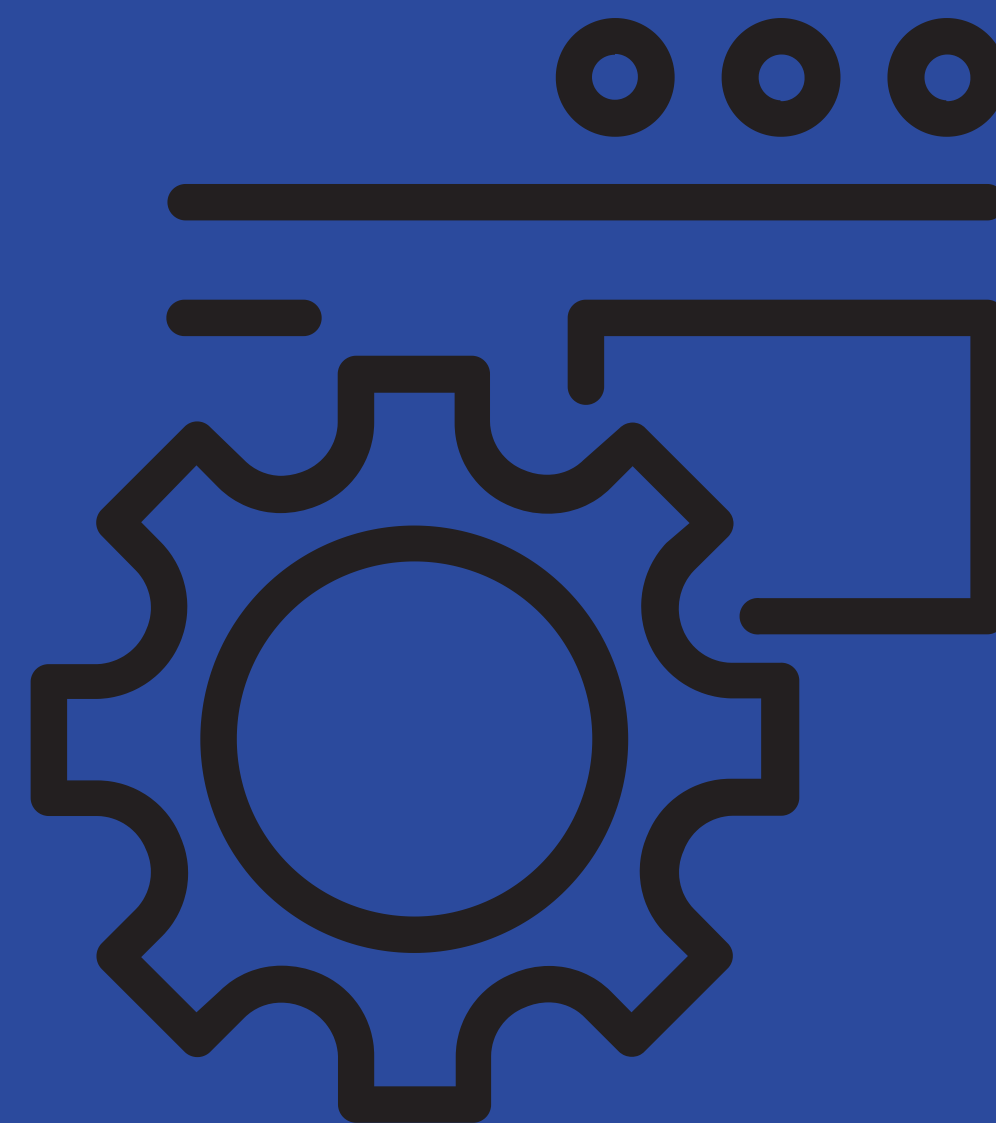
```
CALL AddPhim('FF01', 'Fast and Furious 10', 'OF11', 'Louis Leterrier', 'AC01', 'USA', '2023-05-19 00:00:00', '2023-06-19 00:00:00', 0);
```

- KẾT QUẢ

```
ERROR: Phim đã tồn tại
CONTEXT: PL/pgSQL function addphim(character varying,character varying,character varying,character varying,character varying,character varying,timestamp without time zone,timestamp without time zone,money) line 17 at RAISE
```

IV

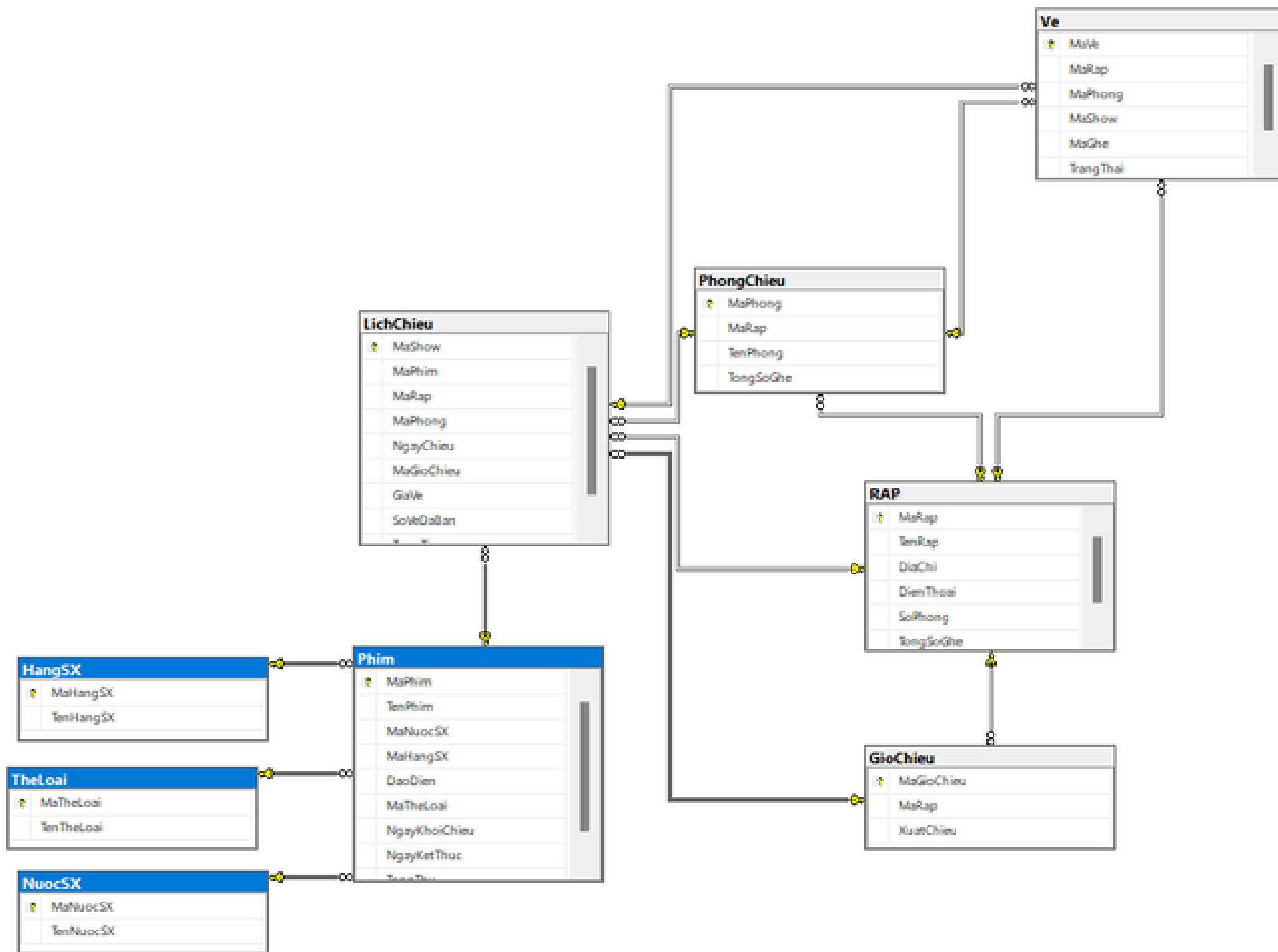
DEMO_CSDL RẠP
CHIẾU PHIM



1. MÔ TẢ CSDL

- **PHIM**(**MaPhim**, TenPhim, MaNuocSX, MaHangSX, MaTheLoai, NgayKhoiChieu, NgayKetThuc, TongThu)
- **HangSX** (**MaHangSX**, TenHangSX)
- **NuocSX** (**MaNuocSX**, TenNuocSX)
- **TheLoai** (**MaTheLoai**, TenTheLoai)
- **Rap** (**MaRap**, TenRap, DiaChi, DienThoai, SoPhong, TongSoGhe)
- **GioChieu** (**MaGioChieu**, MaRap, GioChieu)
- **PhongChieu** (**MaPhong**, MaRap, TenPhong, TongSoGhe)
- **LichChieu** (**MaShow**, MaPhim, MaRap, MaPhong, NgayChieu, MaGioChieu, GiaVe, SoVeDaBan, TongTien)
- **Ve** (**MaVe**, MaRap, MaPhong, MaGhe, MaShow, TrangThai)

2. SƠ ĐỒ LIÊN KẾT GIỮA CÁC BẢNG



3. VIDEO DEMO CSDL

- Tạo bảng, khóa chính, khóa ngoại
- Thêm dữ liệu
- TRIGGER
 - Cập nhập tổng doanh thu và số vé đã bán
 - Cập nhập số phòng và số ghế của 1 rạp khi thêm, xóa, sửa
- FUNCTION
 - In thông tin vé đã được bán
 - Nhập vào tên phim, in thông tin xuất chiếu và số vé còn trống tại các rạp
- VIEW : Tạo bảng lưu thông tin báo cáo doanh thu theo tháng năm 2022 theo từng rạp
- INDEX : Tạo index cho các bảng
- Phân quyền : Tạo user khách hàng, nhân viên, quản lý giám đốc

3. VIDEO DEMO CSDL

V ĐÁNH GIÁ VÀ KẾT LUẬN



1. ƯU ĐIỂM

- Tính bảo mật cao
- Truy vấn mạnh mẽ
- Khả năng mở rộng tốt
- Có khả năng chạy trên web, ứng dụng di động
- Lưu lại nhật kí, hình thành cơ sở dữ liệu hỗ trợ sửa lỗi

2. NHƯỢC ĐIỂM

- Khó học và sử dụng đối với người mới
- Hiệu suất hoạt động thấp hơn so với một số hệ quản trị CSDL nổi bật khác như MySQL, Oracle
- Yêu cầu tài nguyên máy tính cao hơn so với các hệ quản trị CSDL khác
- Các tính năng mới ít được cập nhật thường xuyên

3. SO SÁNH VỚI SQL SERVER

Đặc điểm	PostgreSQL	SQL Sever
Mô hình	Là hệ thống quản lý cơ sở dữ liệu quan hệ mã nguồn mở và miễn phí, được duy trì bởi PostgreSQL Global Development Group và cộng đồng phát triển của nó	Là hệ thống quản lý cơ sở dữ liệu phục vụ cho thương mại, được xây dựng và bảo trì bởi Microsoft
Hệ điều hành máy chủ được hỗ trợ	Được sử dụng khá phổ biến. Nó có sẵn rộng rãi trên nhiều hệ điều hành: FreeBSD, HP-UX, Linux, NetBSD, OpenBSD, OS X, Solaris, Unix, Windows	Được giới hạn trong Windows và gần đây là Linux
Thủ tục lưu trữ	Sử dụng bằng ngôn ngữ độc quyền PL / pgSQL hoặc với các ngôn ngữ phổ biến như Perl, Python, Tcl, v.v	Sử dụng ngôn ngữ Transact SQL và .NET
Phương pháp phân vùng	Phân vùng khai báo (theo phạm vi hoặc theo danh sách) kể từ PostgreSQL 10.0	Các bảng có thể được phân phối trên một số tệp (phân vùng ngang); bảo vệ thông qua liên đoàn
Cài đặt và cập nhật	Đôi khi dễ dàng như gõ một lệnh:sudo apt-get install postgresql	Chậm và liên quan đến tải xuống lớn và quá trình dài phức tạp
Tình trạng cập nhật	Thường xuyên cập nhật phiên bản mới	Phát hành phiên bản mới sau một vài năm

4. NHẬN XÉT

- PostgreSQL: Hệ quản trị cơ sở dữ liệu mạnh mẽ và linh hoạt, dựa trên nguyên tắc ACID.
- Sử dụng đa luồng và có khả năng mở rộng.
- Phù hợp với ứng dụng từ quy mô nhỏ đến lớn.
- Đa dạng loại hình công ty sử dụng PostgreSQL.
- PostgreSQL có thể lưu trữ và quản lý các loại dữ liệu khác nhau.

5. KẾT LUẬN

- PostgreSQL là một giải pháp quản lý cơ sở dữ liệu mã nguồn mở, được sử dụng phổ biến và mạnh mẽ.
- Cung cấp nhiều tính năng và chức năng cho người dùng, bao gồm tính linh hoạt, khả năng mở rộng cao và nhiều loại dữ liệu được hỗ trợ.
- Hỗ trợ khối dữ liệu lớn và các tính năng bảo mật mạnh mẽ như kiểm tra năng lực và kiểm soát truy cập
- Có khả năng tùy chỉnh để phù hợp với nhu cầu của người dùng, tuy nhiên, yêu cầu tài nguyên cao hơn và khó cấu hình hơn các hệ thống cơ sở dữ liệu khác

Tóm lại, PostgreSQL là một lựa chọn phù hợp cho các dự án quản trị cơ sở dữ liệu lớn và phức tạp

CẢM ƠN THẦY VÀ CÁC BẠN
ĐÃ LẮNG NGHE.