

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC VÀ KỸ THUẬT THÔNG TIN



BÁO CÁO ĐỒ ÁN
MÔN QUẢN LÝ THÔNG TIN
TÌM HIỂU HỆ QUẢN TRỊ CSDL POSTGRESQL

Sinh viên thực hiện:

Phạm Nguyên Nam – 20521634

Nguyễn Phúc Khang – 21522194

Giảng viên:

TS – Nguyễn Gia Tuấn Anh

CN - Trần Quốc Khánh

Thành phố Hồ Chí Minh, tháng 6 năm 2023

BÁO CÁO TÓM TẮT

1. Danh sách thành viên

MSSV	Họ tên	Ghi chú
20521634	Phạm Nguyên Nam	
21522194	Nguyễn Phúc Khang	

2. Phân công công việc

MSSV	Họ tên	Nội dung được phân công
200521634	Phạm Nguyên Nam	Chương 1 Chương 2 Chương 3 Chương 5 File thuyết trình Thuyết trình
21522194	Nguyễn Phúc Khang	Lời mở đầu Chương 3 Chương 4 Chương 5 File thuyết trình Thuyết trình

MỤC LỤC

LỜI MỞ ĐẦU	6
CHƯƠNG 1 : TỔNG QUAN VỀ POSTGRESQL	7
1. PostgreSQL là gì ?	7
2. Đặc điểm nổi bật của PostgreSQL.....	7
CHƯƠNG 2 : CÀI ĐẶT VÀ CẤU HÌNH POSTGRESQL.....	7
CHƯƠNG 3 : CÁC LỆNH TRONG POSTGRESQL	12
1. Các chức năng cơ bản (truy vấn, lưu trữ dữ liệu, ...).....	12
1.1. Thao tác với bảng	12
1.1.1. Tạo bảng	12
1.1.2. Sửa bảng.....	13
1.1.3. Xóa bảng.....	13
1.2. Truy xuất dữ liệu	14
1.2.1. Truy xuất dữ liệu từ một bảng	14
1.2.2. Truy vấn lựa chọn một số cột.....	14
1.2.3. Truy vấn sử dụng mệnh đề Where	14
1.2.4. Truy vấn sử dụng mệnh đề Order By	14
1.3. Sử dụng các toán tử trong SQL	14
1.3.1. Toán tử so sánh	14
1.3.2. Toán tử logic	15
1.4. Truy xuất dữ liệu từ nhiều bảng.....	15
1.4.1. Inner join:	15
1.4.2. Left join.....	15
1.5. Thao tác với dữ liệu.....	15
1.5.1. SELECT	15
1.5.2. INSERT	15
1.5.3. UPDATE	16
1.5.4. DELETE	16
2. Các chức năng nâng cao (phân quyền, lưu trữ, backup, restore, chức năng phân tán dữ liệu, bảo mật, ...) mà hệ quản trị đó hỗ trợ.	16
2.1. CONSTRAINT	16
2.1.1. Primary key (khóa chính)	16

2.1.2. Foreign key (khóa ngoại).....	16
2.1.3. Unique (ràng buộc khóa duy nhất)	17
2.1.4. Check.....	17
2.1.5. Not NULL.....	18
2.2.VIEWS	18
2.2.1. REGULAR VIEW	19
2.2.2. UPDATABLE VIEW	19
2.2.3. MATERIALIZED VIEW	19
2.2.4. RECURSIVE VIEW	20
2.2.5. DROP VIEW	20
2.3.INDEX	20
2.3.1. CREATE INDEX	21
2.3.2. DROP INDEX.....	22
2.4.TRIGGER	22
2.5.Chức thực và phân quyền.....	23
2.5.1. TẠO USER	23
2.5.2. PHÂN QUYỀN	23
2.6 BACKUP VÀ RESTORE.....	24
2.6.1. Backup.....	24
2.6.2. Restore	27
2.7.IMPORT VÀ EXPORT	28
2.7.1. Import	28
2.7.2. Export.....	30
2.8 Stored Procedure	32
2.9 FUNCTION.....	34
CHƯƠNG 4 : CSDL QUẢN LÝ RẠP CHIẾU PHIM	35
1. Mô tả CSDL	35
2. Thuộc tính của các thành phần	36
3. Sơ đồ liên kết quan hệ của các bảng	38
4. Video quay demo CSDL quản lý rạp chiếu phim.....	39
CHƯƠNG 5 : ĐÁNH GIÁ VÀ KẾT LUẬN	39
1. Ưu và nhược điểm.....	39
a. Ưu điểm	39

b. Nhược điểm	39
2. So sánh PostgreSQL và SQL Server	39
3. Nhận xét :	40
4. Kết luận	40
TÀI LIỆU THAM KHẢO.....	41

PHỤ LỤC HÌNH ẢNH

Hình 1.1 Logo postgresSQL.....	7
Hình 2.1 Hình ảnh tải xuống trên trang chủ	8
Hình 2.2 Hình ảnh chạy tệp trên máy tính.....	9
Hình 2.3 Hình ảnh chọn đường dẫn cài đặt.	9
Hình 2.4 Hình ảnh chọn các thành phần để cài đặt vào máy.	10
Hình 2.5 Hình ảnh chọn đường dẫn lưu dữ liệu.	10
Hình 2.6 Hình ảnh thiết lập mật khẩu.....	11
Hình 2.7 Hình ảnh chọn port.	11
Hình 2.8 Màn hình hoàn thành cài đặt trên máy.....	12
Hình 3.1 Ảnh minh họa cách tạo bảng.	13
Hình 3.2 Ảnh minh họa cách thêm cột vào bảng.....	13
Hình 3.3 Ảnh minh họa sửa cột trong bảng.	13
Hình 3.4 Ảnh minh họa xóa cột trong bảng.....	13
Hình 3.5 Ảnh minh họa xóa bảng.	13
Hình 3.6 Ảnh minh họa truy vấn tất cả dữ liệu từ một bảng.	14
Hình 3.7 Ảnh minh họa truy vấn một cột trong một bảng.....	14
Hình 3.8 Ảnh minh họa truy vấn dùng mệnh đề Where.	14
Hình 3.9 Ảnh minh họa sử dụng mệnh đề ORDER BY.	14
Hình 3.10 Hình ảnh minh họa sử dụng toán tử so sánh.....	15
Hình 3.11 Ảnh minh họa sử dụng toán tử so sánh.	15
Hình 3.12 Ảnh minh họa truy xuất nhiều bảng với INNER JOIN.	15
Hình 3.13 Ảnh minh họa truy xuất nhiều bảng với LEFT JOIN.	15
Hình 3.14 Ảnh minh họa sử dụng SELECT để truy xuất dữ liệu.	15
Hình 3.15 Ảnh minh họa sử dụng INSERT để thêm dữ liệu.	16
Hình 3.16 Ảnh minh họa sử dụng UPDATE để cập nhật, sửa đổi dữ liệu.	16
Hình 3.17 Ảnh minh họa sử dụng DELETE để xóa dữ liệu.	16
Hình 3.18 Ảnh minh họa tạo khóa chính.....	16
Hình 3.19 Ảnh minh họa tạo khóa ngoại.....	17
Hình 3.20 Ảnh minh họa tạo ràng buộc khóa duy nhất.	17
Hình 3.21 Minh họa kiểm tra ràng buộc khóa duy nhất.	17
Hình 3.22 Ảnh minh họa tạo ràng buộc CHECK.	17
Hình 3.23 Ảnh minh họa tạo ràng buộc CHECK.	18
Hình 3.24. Ảnh minh họa tạo ràng buộc Not NULL.	18
Hình 3.25 Chạy thử minh họa chạy ràng buộc Not NULL.....	18
Hình 3.26 Ảnh minh họa tạo bảng View.....	19
Hình 3.27 Truy xuất bảng View.	19
Hình 3.28 Kiểm tra View.	19
Hình 3.29 Materialized View.	20
Hình 3.30. Recursive View.	20
Hình 3.31 Drop View.....	20
Hình 3.32 Create index.	21
Hình 3.33 Create B-Tree index.....	21
Hình 3.34 Create Hash index.....	21
Hình 3.35. Create GiST index.	21

Hình 3.36 Create GIN index.....	21
Hình 3.37 Create SP-Gist index.....	21
Hình 3.38 Create BRIN index.	22
Hình 3.39 Tạo giao diện.	22
Hình 3.40 Drop Index.....	22
Hình 3.41 Trigger.....	23
Hình 3.42 Ảnh minh họa tạo User.	23
Hình 3.43 Ảnh minh họa tạo đặc quyền.	24
Hình 3.44 Ảnh minh họa Backup.	25
Hình 3.45 Ảnh minh họa chọn đường dẫn.....	25
Hình 3.46 Ảnh minh họa chọn định dạng kiểu, vị trí và tên cho tệp.....	26
Hình 3.47 Ảnh minh họa khi thành công.....	26
Hình 3.48 Ảnh minh họa file cần kiểm tra.	26
Hình 3.49 Ảnh minh họa chọn restore.....	27
Hình 3.50 Ảnh minh họa chọn file trong đường dẫn.....	27
Hình 3.51 Ảnh minh họa restore.....	28
Hình 3.52 Ảnh minh họa restore thành công.....	28
Hình 3.53 Ảnh minh họa chọn Import Database.	28
Hình 3.54 Ảnh minh họa chưa có số liệu trong bảng.	29
Hình 3.55 Ảnh minh họa chọn file và đường dẫn.....	29
Hình 3.56 Ảnh minh họa chọn thông tin cần nhập.....	29
Hình 3.57. Ảnh minh họa Import thành công.....	30
Hình 3.58 Ảnh minh họa Import thành công.....	30
Hình 3.59 Ảnh minh họa chọn Export Database.	30
Hình 3.60 Ảnh minh họa chọn file trong đường dẫn.....	31
Hình 3.61 Ảnh minh họa chọn thông tin cần xuất.....	31
Hình 3.62 Ảnh minh họa xuất thành công.....	31
Hình 3.63 Ảnh minh họa thông tin được xuất vào file Doc4	32
Hình 3.64 Ảnh minh họa cú pháp Stored Procedure	32
Hình 3.65 Ảnh minh họa ví dụ Store Procedure.....	33
Hình 3.66 Hàm gọi ví dụ Store Procedure.....	33
Hình 3.67 Bảng trước khi thực hiện hàm	33
Hình 3.68 Bảng sau khi thực hiện hàm.....	33
Hình 3.69 Cú pháp sử dụng FUNCTION.....	34
Hình 3.70 Hàm gọi ví dụ gọi FUNCTION	34
Hình 3.71 Hàm xóa FUNCTION.....	34
Hình 3.72 Dữ liệu được có trong bảng nhân viên.....	34
Hình 3.73 Hàm FUNCTION đếm số lượng nhân viên.....	35
Hình 3.74 Kết quả số lượng nhân viên trả về	35
Hình 4.1 Bảng quan hệ giữa các bảng	38

LỜI MỞ ĐẦU

Trong thời đại công nghệ thông tin phát triển vượt bậc, dữ liệu đã trở thành tài nguyên quý giá và quyết định đến sự thành công của nhiều tổ chức và doanh nghiệp. Để quản lý và tận dụng tối đa giá trị của dữ liệu, hệ quản trị cơ sở dữ liệu đóng vai trò quan trọng. Trong đề án này, nhóm em đã tiến hành nghiên cứu về một trong những hệ quản trị cơ sở dữ liệu phổ biến và mạnh mẽ - PostgreSQL.

Lời mở đầu này nhằm giới thiệu đến mọi người về mục tiêu, phạm vi và phương pháp tiếp cận của đề án. Nhóm đã tiến hành một quá trình nghiên cứu chi tiết và thực tế về PostgreSQL, nhằm tìm hiểu về kiến trúc, tính năng, ưu điểm và nhược điểm của hệ quản trị cơ sở dữ liệu này.

Nhóm em bắt đầu bằng việc đưa ra khái quát về PostgreSQL và đặt ra câu hỏi nghiên cứu. Tiếp theo, nhóm đã tìm hiểu về kiến trúc và các thành phần cơ bản của PostgreSQL, bao gồm quản lý lưu trữ, truy vấn dữ liệu, quản lý người dùng và bảo mật.

Sau đó, đã tiến hành các thử nghiệm và phân tích về hiệu suất và khả năng mở rộng của PostgreSQL. Nhóm đã sử dụng tải dữ liệu lớn và thực thi các câu truy vấn phức tạp để đánh giá hiệu suất của hệ thống.

Qua quá trình nghiên cứu và thực hiện, chúng em đã thu thập được các kết quả và phân tích chi tiết. Đề án này sẽ cung cấp một cái nhìn toàn diện về PostgreSQL, từ khía cạnh kỹ thuật đến khả năng ứng dụng thực tế. Chúng em hy vọng rằng đề án này sẽ đóng góp vào việc hiểu rõ hơn về hệ quản trị cơ sở dữ liệu PostgreSQL và mang lại giá trị cho mọi người trong việc quản lý và tối ưu hóa cơ sở dữ liệu của họ.

CHƯƠNG 1 : TỔNG QUAN VỀ POSTGRESQL

1. PostgreSQL là gì ?

- **PostgreSQL** là một hệ quản trị cơ sở dữ liệu quan hệ và đối tượng dựa trên POSTGRES bản 4.2, được khoa điện toán của đại học California tại Berkeley phát triển vào năm 1986. POSTGRES mở đường cho nhiều khái niệm quan trọng mà các hệ quản trị dữ liệu thương mại rất lâu sau mới có.
- Thời gian đầu, PostgreSQL chỉ có thể chạy trên UNIX và một số hệ nền tảng tương tự khác. Sau một thời gian, PostgreSQL đã có thể chạy trên nhiều nền tảng phổ biến như Windows, macOS và Solaris



Hình 1.1 Logo postgresQL

2. Đặc điểm nổi bật của PostgreSQL

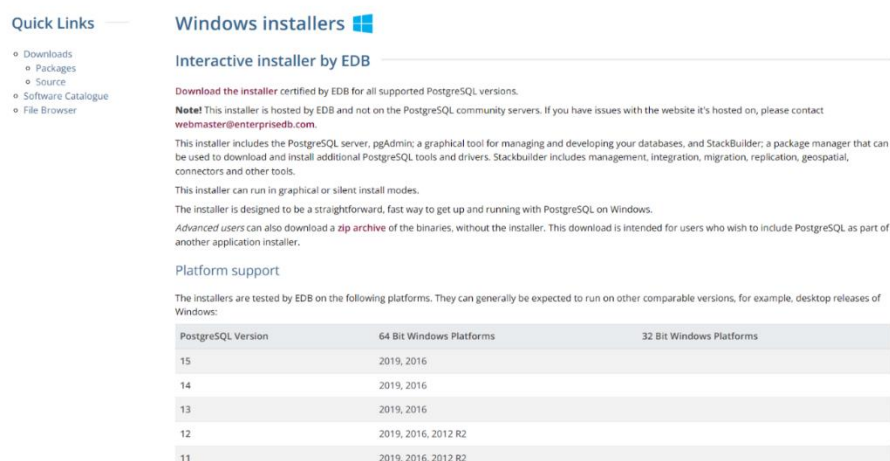
- **PostgreSQL** chứa nhiều điểm nổi bật của các hệ CSDL cũ và có thêm một vài tính năng mới, hỗ trợ cho nhiều dự án và thuận tiện cho người sử dụng:
 - **Chức năng bảo mật:** hỗ trợ xây dựng hàng rào bảo mật, xác thực mạnh mẽ
 - **Tính toàn vẹn dữ liệu :** được đảm bảo toàn vẹn nhờ vào các khóa như Primary Keys, Foreign Keys, UNIQUE, NOT NULL, khóa hàm số/ Explicit Locks, Khóa khuyến nghị / Advisory Locks, ...
 - Hỗ trợ Native Microsoft Window Server
 - Phục hồi, chỉnh sửa từng phần, từng thời điểm
 - Khả năng mở rộng
 - Hỗ trợ nhiều ngôn ngữ : PL/PGSQL, Ruby, Perl, Python ...
 - Hỗ trợ nhiều kiểu dữ liệu : loại hình học, địa chỉ mạng (cidr, inet, macaddr), hàm, các kiểu dữ liệu tùy chỉnh ...

CHƯƠNG 2 : CÀI ĐẶT VÀ CẤU HÌNH POSTGRESQL

1. Cài đặt và cấu hình POSTGRESQL.

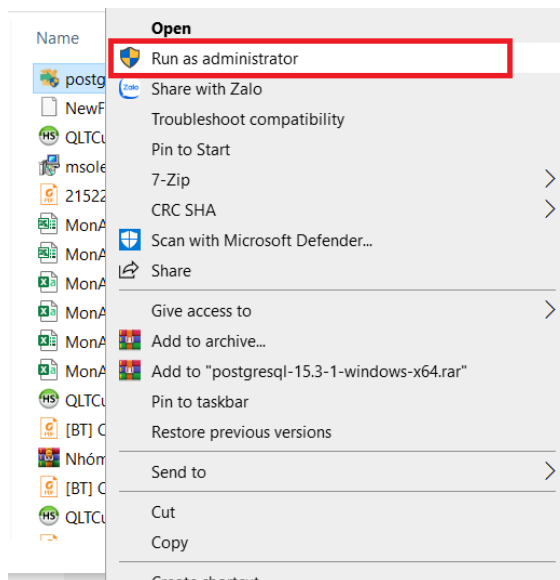
Bước 1: Tải xuống PostgreSQL

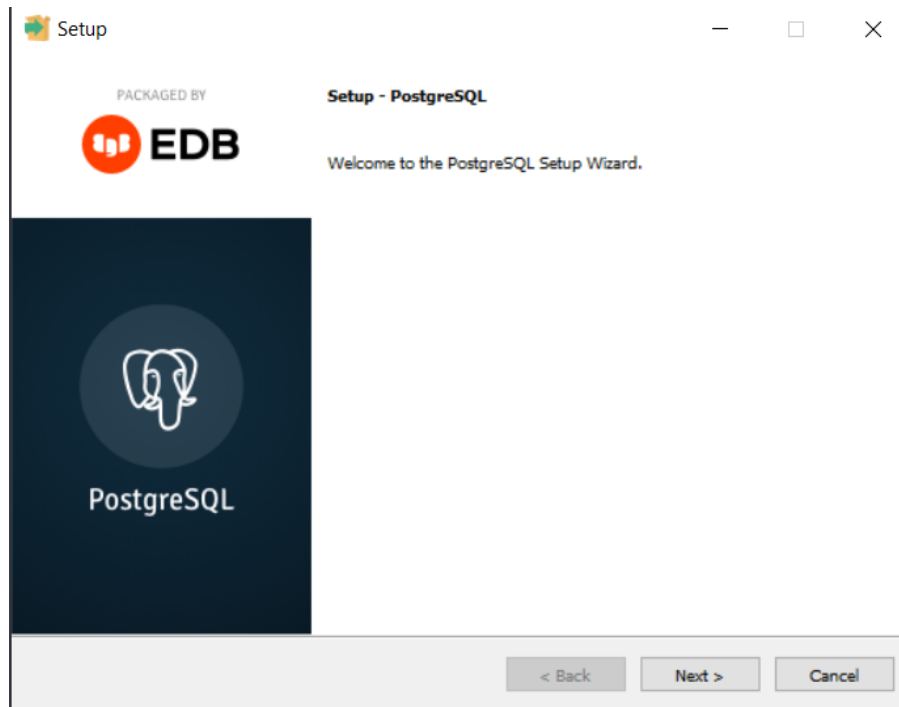
- Truy cập vào trang web chính thức của PostgreSQL tại địa chỉ: <https://www.postgresql.org/download/windows/>
- Chọn phiên bản phù hợp với hệ điều hành của bạn (32-bit hoặc 64-bit).
- Nhấp vào liên kết tải xuống phù hợp để tải xuống bản cài đặt.



Hình 2.1 Hình ảnh tải xuống trên trang chủ

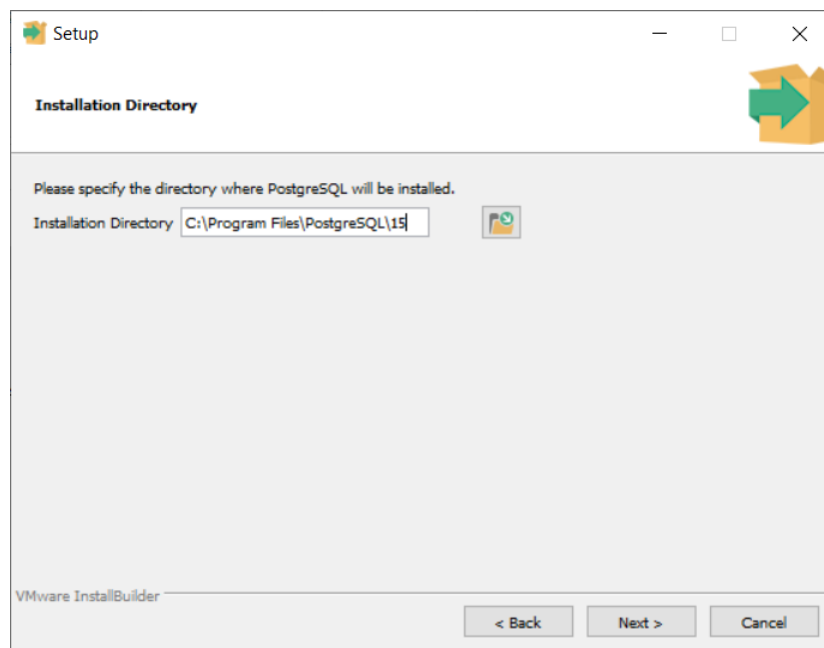
Bước 2: Cài đặt PostgreSQL ,chạy tệp cài đặt đã tải xuống.





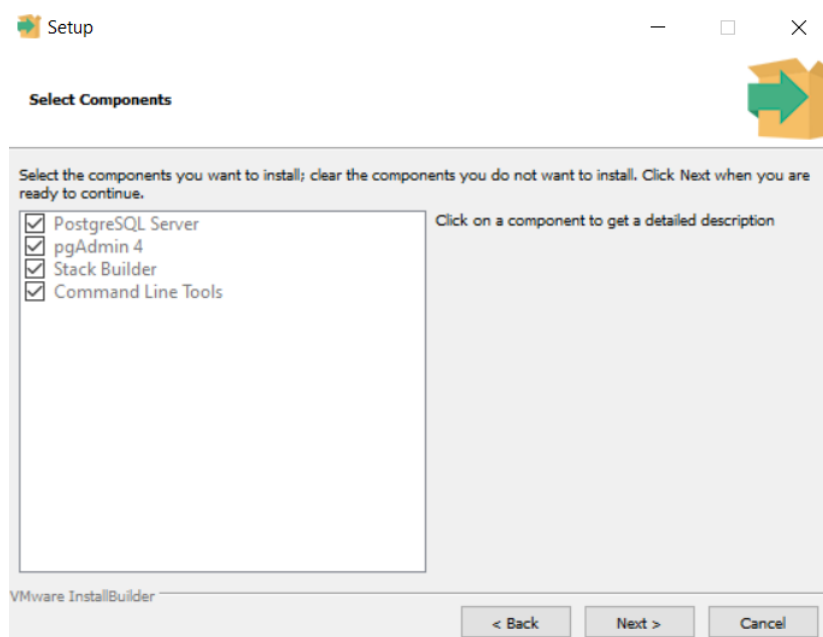
Hình 2.2 Hình ảnh chạy tệp trên máy tính

Chọn đường dẫn cài đặt cho PostgreSQL hoặc để mặc định và nhấp vào nút "Next".



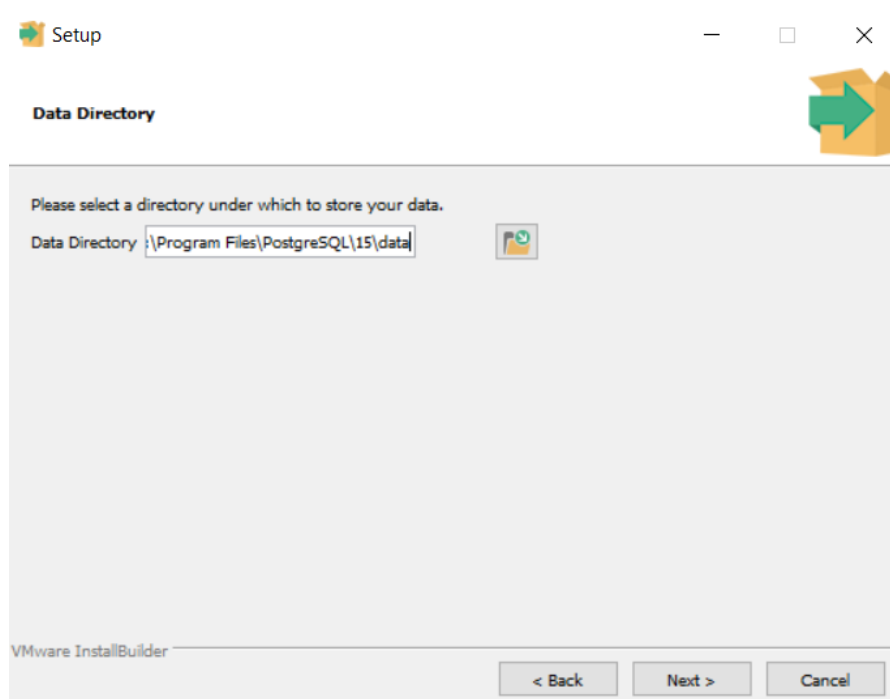
Hình 2.3 Hình ảnh chọn đường dẫn cài đặt.

Chọn các thành phần bạn muốn cài đặt. Mặc định, tất cả các thành phần đã chọn sẽ được cài đặt. Nhấp vào nút "Next" để tiếp tục.



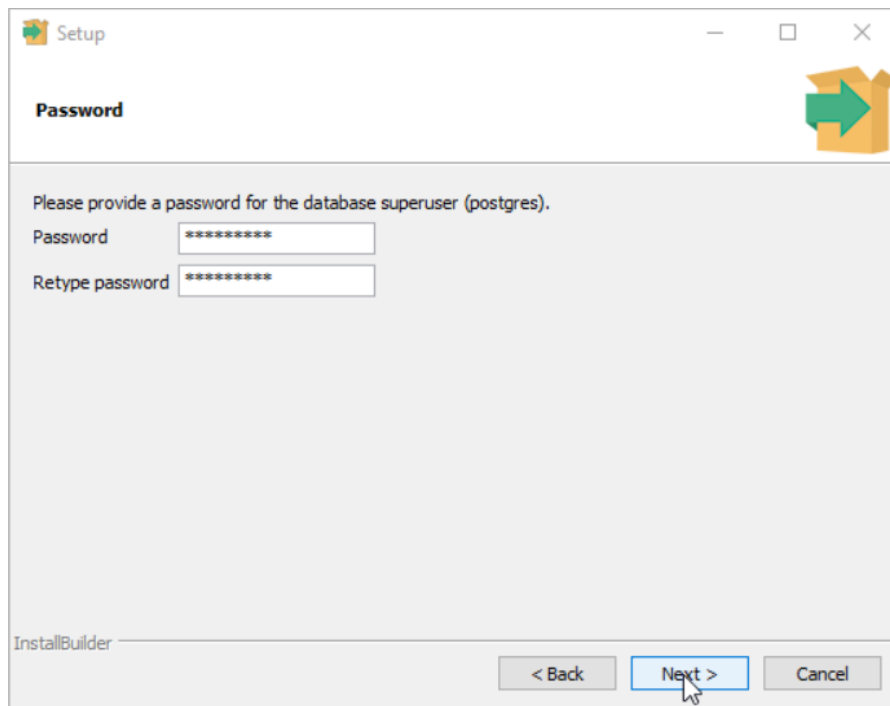
Hình 2.4 Hình ảnh chọn các thành phần để cài đặt vào máy.

Chọn thư mục dữ liệu (data directory) cho PostgreSQL. Đây là nơi PostgreSQL sẽ lưu trữ cơ sở dữ liệu của bạn. Bạn có thể để mặc định hoặc chọn một thư mục khác và nhấp vào nút "Next".



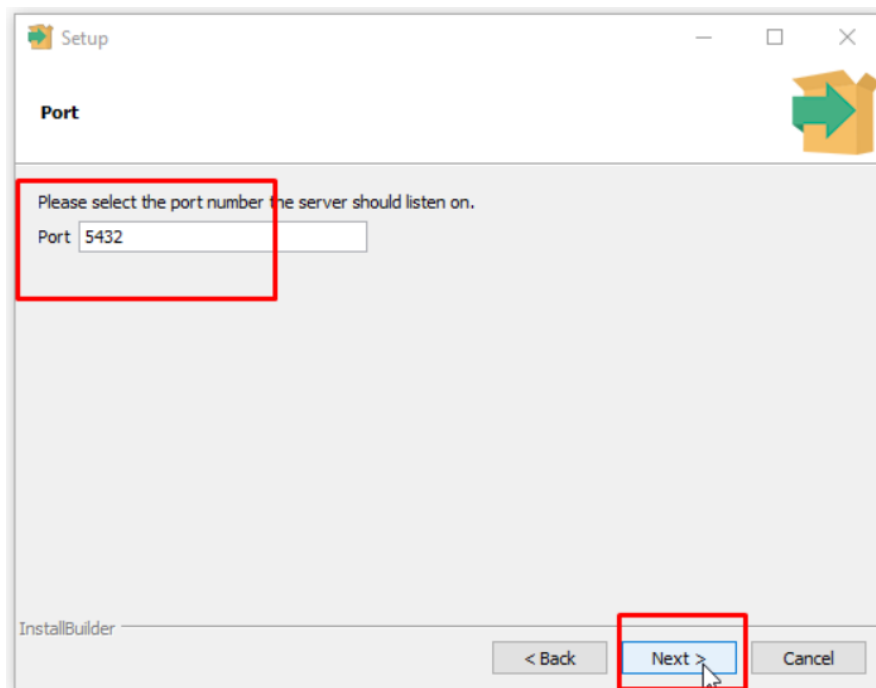
Hình 2.5 Hình ảnh chọn đường dẫn lưu dữ liệu.

Nhập mật khẩu cho người dùng postgres. Đây là mật khẩu cho tài khoản quản trị của PostgreSQL. Ghi nhớ mật khẩu này vì bạn sẽ cần nó để đăng nhập sau này. Nhấp vào nút "Next" để tiếp tục.



Hình 2.6 Hình ảnh thiết lập mật khẩu.

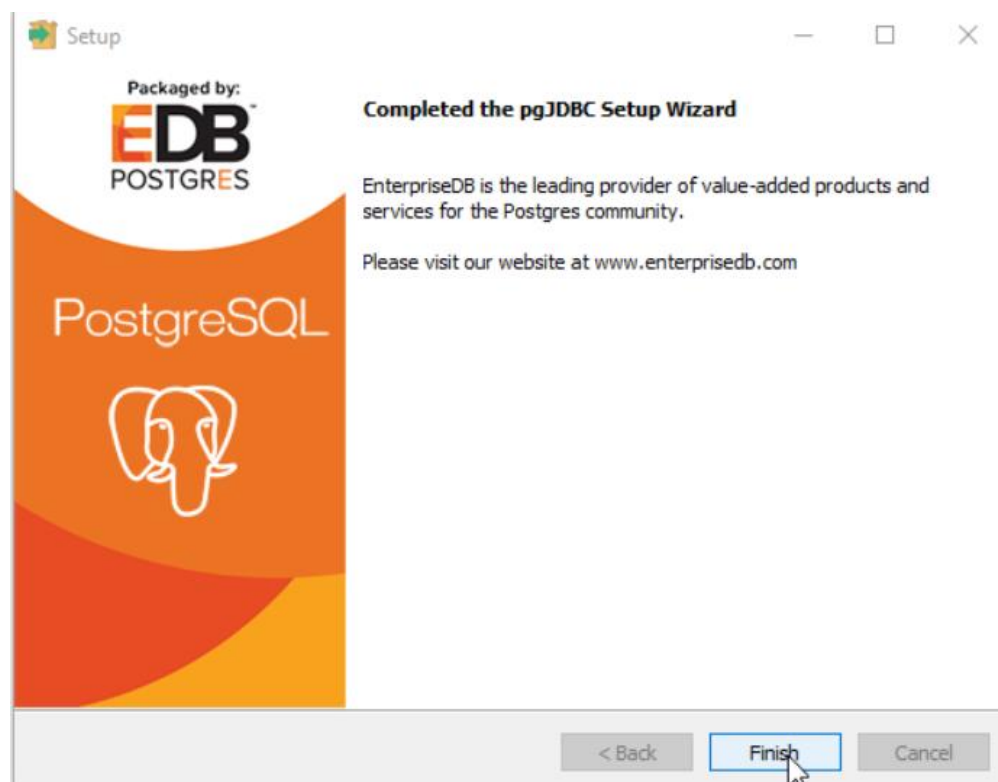
Chọn cổng mặc định cho PostgreSQL hoặc để giá trị mặc định và nhấp vào nút "Next".



Hình 2.7 Hình ảnh chọn port.

Kiểm tra lại các tùy chọn đã chọn và nhấp vào nút "Next".

Chờ quá trình cài đặt hoàn tất và nhấp vào nút "Finish" để kết thúc.



Hình 2.8 Màn hình hoàn thành cài đặt trên máy.

CHƯƠNG 3 : CÁC LỆNH TRONG POSTGRESQL

1. Các chức năng cơ bản (truy vấn, lưu trữ dữ liệu, ...)

1.1. Thao tác với bảng

1.1.1. Tạo bảng

- *Cú pháp:* CREATE TABLE

- **Chức năng:** Khởi tạo một bảng rỗng mới trong CSDL. Bảng này sẽ thuộc sở hữu của user đã viết ra câu lệnh này.

Ví dụ: Câu lệnh trên tạo ra 1 bảng có tên là KHACHHANG và bảng đó chứa 4 cột MAKH, TENKH, DIACHI và LOAIKH.

```
1 CREATE TABLE KHACHHANG (
2 MAKH CHAR(4) NOT NULL,
3 TENKH VARCHAR(40),
4 DIACHI VARCHAR(40),
5 LOAIKH VARCHAR(30)
6 );
```

Hình 3.1 Ảnh minh họa cách tạo bảng.

1.1.2. Sửa bảng

- Thêm cột:
 - Cú pháp: ALTER TABLE tên_bảng ADD tên_cột loại_dữ_liệu;
 - Chức năng: Để thêm một cột vào trong một bảng hiện có.

Ví dụ: Câu lệnh SQL dưới đây sẽ thêm một cột có tên là SĐT vào bảng KHACHHANG:

```
9 ALTER TABLE KHACHHANG ADD SĐT CHAR(10);
10
```

Hình 3.2 Ảnh minh họa cách thêm cột vào bảng.

- Sửa cột:
 - Cú pháp: ALTER TABLE tên_bảng ALTER COLUMN tên_cột loại_dữ_liệu;
 - Chức năng: Để thay đổi loại dữ liệu của một cột trong một bảng dưới đây. **Ví dụ:** Câu lệnh SQL dưới đây sẽ chỉnh sửa cột có tên là SĐT thành CHAR(20).

```
11 ALTER TABLE KHACHHANG ALTER COLUMN SĐT CHAR(20);
12
```

Hình 3.3 Ảnh minh họa sửa cột trong bảng.

- Xóa cột:
 - Cú pháp: ALTER TABLE tên_bảng DROP COLUMN tên_cột;
 - Chức năng: Để xóa một cột trong một bảng (lưu ý rằng một số hệ thống cơ sở dữ liệu không cho phép xóa một cột).

Ví dụ: Câu lệnh SQL dưới đây sẽ xóa cột SĐT khỏi bảng KHACHHANG.

```
13 ALTER TABLE KHACHHANG DROP COLUMN SĐT;
14
```

Hình 3.4 Ảnh minh họa xóa cột trong bảng.

1.1.3. Xóa bảng

- Cú pháp: DROP TABLE tên_bảng;
- Chức năng: Dùng để xóa một bảng trong cơ sở dữ liệu. Sau khi xóa, tất cả dữ liệu trong bảng cũng sẽ mất.

Ví dụ: Câu lệnh SQL dưới đây sẽ xóa bảng NHANVIEN cùng dữ liệu trong bảng khỏi cơ sở dữ liệu.

```
14
15 DROP TABLE NHANVIEN;
```

Hình 3.5 Ảnh minh họa xóa bảng.

1.2. Truy xuất dữ liệu

1.2.1. Truy xuất dữ liệu từ một bảng

- *Cú pháp:* `SELECT * FROM <tên TABLE>,...`
- *Chức năng:* Truy vấn lựa chọn tất cả các hàng và cột.

Ví dụ: Câu lệnh SQL dưới đây sẽ lấy tất cả dữ liệu ở KHACHHANG.

```
17 SELECT * FROM KHACHHANG;  
18
```

Hình 3.6 Ảnh minh họa truy vấn tất cả dữ liệu từ một bảng.

1.2.2. Truy vấn lựa chọn một số cột

- *Cú pháp:* `SELECT <Tên table> FROM <tên table1>, <tên table2>,...`
- *Chức năng:* Lấy dữ liệu của một Table trong câu truy vấn có nhiều Table.

Ví dụ: Câu lệnh sau đây sẽ lấy ra MAKH trong bảng KHACHHANG.

```
18  
19 SELECT MAKH FROM KHACHHANG;
```

Hình 3.7 Ảnh minh họa truy vấn một cột trong một bảng.

1.2.3. Truy vấn sử dụng mệnh đề Where

- *Chức năng:* Dùng để trích xuất những bản ghi (dòng dữ liệu) thỏa mãn điều kiện đặt ra.
- *Cú pháp:* `SELECT <Tên table> FROM <tên table1>, <tên table2> WHERE <điều kiện>`

Ví dụ: Câu truy vấn sau đây sẽ lấy MASP, TENSP từ bảng SANPHAM với điều kiện NUOCSX là TRUNG QUOC.

```
19 SELECT MASP, TENSP  
20 FROM SANPHAM  
21 WHERE NUOCSX = 'TRUNG QUOC'
```

Hình 3.8 Ảnh minh họa truy vấn dùng mệnh đề Where.

1.2.4. Truy vấn sử dụng mệnh đề Order By

- *Chức năng:* Cho phép sắp xếp kết quả truy vấn theo cột và có thể sắp xếp kết quả theo chiều: Tăng dần (asc) hoặc giảm dần (desc).
- *Cú pháp:* `ORDER BY <Tên table> ASC/DESC`

Ví dụ: Câu truy vấn này sẽ lấy ra SOHD, TRIGIA từ bảng HOADON trong thời gian hoạt động: 1/2007 với ngày hoạt động xếp tăng dần và trị giá giảm dần.

```
25 SELECT SOHD, TRIGIA  
26 FROM HOADON  
27 WHERE MONTH(NGHD) = 1 AND YEAR(NGHD) = 2007  
28 ORDER BY NGHD ASC, TRIGIA DESC
```

Hình 3.9 Ảnh minh họa sử dụng mệnh đề ORDER BY.

1.3. Sử dụng các toán tử trong SQL

1.3.1. Toán tử so sánh

Ví dụ: Kết quả tra về của lệnh SELECT nói trên là in ra các số hóa đơn, trị giá hóa đơn bán ra trong ngày 1/1/2007 và ngày 2/1/2007.

```
SELECT SOHD, TRIGIA
FROM HOADON
WHERE NGHD >= '1/1/2007' AND NGHD <= '1/2/2007'
```

Hình 3.10 Hình ảnh minh họa sử dụng toán tử so sánh.

1.3.2. Toán tử logic

Ví dụ: Kết quả tra về của lệnh SELECT nói trên là In ra danh sách các sản phẩm (MASP, TENS P) do “Trung Quốc” sản xuất có giá từ 30.000 đến 40.000.

```
4 SELECT MASP, TENS P, NUOCSX
5 FROM SANPHAM
6 WHERE NUOCSX = 'TRUNG QUOC'
7 AND GIA BETWEEN 30000 AND 40000
```

Hình 3.11 Ảnh minh họa sử dụng toán tử so sánh.

1.4. Truy xuất dữ liệu từ nhiều bảng

1.4.1. Inner join:

Ví dụ: Câu lệnh sau sẽ in ra danh sách các sản phẩm (MASP, TENS P) không bán được trong năm 2006.

```
8 SELECT S.MASP, TENS P
9 FROM SANPHAM S
10 WHERE S.MASP NOT IN (SELECT C.MASP
11 FROM CTHD C INNER JOIN HOADON H
12 ON C.SOHD = H.SOHD
13 WHERE YEAR(NGHD) = 2006)
```

Hình 3.12 Ảnh minh họa truy xuất nhiều bảng với INNER JOIN.

1.4.2. Left join

Ví dụ: Câu lệnh sau sẽ in ra sản phẩm (MASP, TENS P) có giá bán cao nhất của mỗi nước

```
14 SELECT B.NUOCSX, MASP, TENS P
15 FROM (SELECT NUOCSX, MAX(GIA) AS MAX
16 FROM SANPHAM
17 GROUP BY NUOCSX) AS B LEFT JOIN SANPHAM S
18 ON S.GIA = B.MAX
19 WHERE B.NUOCSX = S.NUOCSX
```

Hình 3.13 Ảnh minh họa truy xuất nhiều bảng với LEFT JOIN.

1.5. Thao tác với dữ liệu.

1.5.1. SELECT

- Chức năng: SELECT dùng để truy xuất (lấy) dữ liệu từ một bảng trong Database.
- Cú pháp: SELECT column_1, column_2, ..., column_n FROM TABLE_NAME WHERE Condition;

Ví dụ: Câu lệnh sau sẽ truy xuất MAKH, HOTEN những khách hàng loại Vip trong bảng KHACHHANG.

```
20 SELECT MAKH, HOTEN
21 FROM KHACHHANG
22 WHERE LOAIKH = 'Vip'
```

Hình 3.14 Ảnh minh họa sử dụng SELECT để truy xuất dữ liệu.

1.5.2. INSERT

- Chức năng: INSERT dùng để thêm một dòng dữ liệu mới vào một bảng trong database.

- *Cú pháp:* INSERT INTO TABLE_NAME VALUES ("value_1", "value_2", "value_3", ..., "value_n");

Ví dụ: Câu lệnh sau sẽ chèn những giá trị của một nhân viên như mã nhân viên, họ tên,...

```
23 INSERT INTO NHANVIEN VALUES ('NV01', 'NGUYEN NHU NHUT', '927345678', '13/04/2006')
```

Hình 3.15 Ảnh minh họa sử dụng INSERT để thêm dữ liệu.

1.5.3. UPDATE

- *Chức năng:* UPDATE dùng để cập nhật, sửa đổi một dòng dữ liệu mới vào một bảng trong database.
- *Cú pháp:* UPDATE TABLE_NAME SET COLUMN_NAME = "new_value" WHERE <condition>;

Ví dụ: Câu lệnh sau sẽ cập nhật giá sản phẩm của nước Thái Lan sẽ tăng 5%.

```
24 UPDATE SANPHAM SET GIA=GIA+GIA/(100/5) WHERE NUOCSX='Thai Lan'
```

Hình 3.16 Ảnh minh họa sử dụng UPDATE để cập nhật, sửa đổi dữ liệu.

1.5.4. DELETE

- *Chức năng:* DELETE dùng để xóa những dòng dữ liệu trong database.
- *Cú pháp:* DELETE FROM TABLE_NAME WHERE <condition>;

Ví dụ: Câu lệnh sau sẽ xóa mã khách hàng KH02 ra khỏi bảng KHACHHANG

```
25 DELETE FROM KHACHHANG
26 WHERE MAKH = 'KH02'
```

Hình 3.17 Ảnh minh họa sử dụng DELETE để xóa dữ liệu.

2. Các chức năng nâng cao (phân quyền, lưu trữ, backup, restore, chức năng phân tán dữ liệu, bảo mật, ...) mà hệ quản trị đó hỗ trợ.

2.1.CONSTRAINT

2.1.1. Primary key (khóa chính)

- *Chức năng:* Dùng để định danh mỗi một record trong bảng (table) duy nhất của cơ sở dữ liệu.

Ví dụ: Câu lệnh trên tạo lấy cột MAKH làm khóa chính.

```
1 CREATE TABLE KHACHHANG
2 (
3     MAKH CHAR(4) NOT NULL Primary key,
4     TENKH VARCHAR(40),
5     DIACHI VARCHAR(40),
6     LOAIKH VARCHAR(30)
7 );
```

Hình 3.18 Ảnh minh họa tạo khóa chính.

2.1.2. Foreign key (khóa ngoại)

- *Chức năng:* Dùng để tăng tính tham chiếu trong cơ sở dữ liệu, được xem như con trỏ trỏ tới khóa chính của table khác.

Ví dụ: Câu lệnh trên tạo lấy cột MAKH trên bảng HOADON làm khóa ngoại chiếu đến cột MAKH trên bảng KHACHHANG.

```

18 CREATE TABLE HOADON
19 (
20     SOHD CHAR(5) NOT NULL PRIMARY KEY,
21     NGHD DATE,
22     MAKH CHAR(4),
23     KHUYENMAI INT,
24     CONSTRAINT FK_HD_KH FOREIGN KEY (MAKH) REFERENCES KHACHHANG (MAKH)
25 );

```

Hình 3.19 Ảnh minh họa tạo khóa ngoại.

2.1.3. Unique (ràng buộc khóa duy nhất)

- **Chức năng:** Dùng để đảm bảo rằng chỉ các giá trị duy nhất được nhập vào trong cột hoặc một tập hợp các cột.

Ví dụ: Câu lệnh trên lấy cột EMAIL làm ràng buộc khóa duy nhất.

```

1 CREATE TABLE KHACHHANG
2 (
3     MAKH CHAR(4) NOT NULL Primary key,
4     TENKH VARCHAR(40),
5     DIACHI VARCHAR(40),
6     LOAIKH VARCHAR(30),
7 );
8 ALTER TABLE KHACHHANG ADD COLUMN EMAIL VARCHAR(100);
9 ALTER TABLE KHACHHANG ADD UNIQUE (EMAIL)

```

Hình 3.20 Ảnh minh họa tạo ràng buộc khóa duy nhất.

Kiểm tra: Trong trường hợp này, khi ta để Email là Unique thì khi Insert 2 giá trị có 2 địa chỉ email giống nhau thì hệ thống sẽ báo lỗi như dưới.

```

38 INSERT INTO KHACHHANG VALUES ('KH01','Liz Kim Cuong ','Ha Noi','Vang lai', 'hihi@gmail.com')
39 INSERT INTO KHACHHANG VALUES ('KH02','Ivone Dieu','Da Nang','Thuong xuyen','hihi@gmail.com')
40 INSERT INTO KHACHHANG VALUES ('KH03','Emma Nhat Khanh','TP.HCM','Vang lai','emma@gmail.com')
41

```

Data Output Explain Messages Notifications

ERROR: duplicate key value violates unique constraint "khachhang_email_key"
 DETAIL: Key (email)=(hihi@gmail.com) already exists.
 SQL state: 23505

Hình 3.21 Minh họa kiểm tra ràng buộc khóa duy nhất.

2.1.4. Check

- **Chức năng:** Dùng để giới hạn phạm vi giá trị có thể đặt vào trong một cột.

Ví dụ: Câu lệnh trên tạo ràng buộc CHECK kiểm tra SOLUONG trong bảng CTHD phải ≥ 0 .

```

28 CREATE TABLE CTHD
29 (
30     SOHD CHAR(5) NOT NULL,
31     MALC CHAR(4) NOT NULL,
32     SOLUONG INT,
33     PRIMARY KEY (SOHD, MALC),
34     CONSTRAINT FK_CTHD_HD FOREIGN KEY (SOHD) REFERENCES HOADON (SOHD),
35     CONSTRAINT FK_CTHD_LC FOREIGN KEY (MALC) REFERENCES LOAICAY (MALC)
36 );
37 ALTER TABLE CTHD ADD CONSTRAINT CHECK_SL CHECK (SOLUONG >= 0)

```

Hình 3.22 Ảnh minh họa tạo ràng buộc CHECK.

Kiểm tra: Trong trường hợp này, SOLUONG trong bảng CTHD được đặt điều kiện ≥ 0 . Nếu dữ liệu nhập vào số lượng < 0 , hệ thống sẽ báo lỗi.

```

54 INSERT INTO CTHD VALUES ('00001','LC01','1')
55 INSERT INTO CTHD VALUES ('00001','LC02','2')
56 INSERT INTO CTHD VALUES ('00003','LC03','5')
57 INSERT INTO CTHD VALUES ('00003','LC03','-1')

```

Data Output Explain Messages Notifications

ERROR: new row for relation "cthd" violates check constraint "check_sl"
DETAIL: Failing row contains (00003, LC03, -1).
SQL state: 23514

Hình 3.23 Ảnh minh họa tạo ràng buộc CHECK.

2.1.5. Not NULL

- **Chức năng:** Là ràng buộc không cho phép một cột chứa các giá trị NULL.

Ví dụ: Câu lệnh trên tạo ràng buộc Not NULL trên cột XUATXU của bảng LOAICAY để kiểm tra không cho XUATXU có giá trị NULL.

```

11 CREATE TABLE LOAICAY
12 (
13     MALC CHAR(4) NOT NULL PRIMARY KEY,
14     TENLC VARCHAR(40),
15     XUATXU VARCHAR(10),
16     GIA MONEY
17 );
18 ALTER TABLE LOAICAY ALTER COLUMN XUATXU SET NOT NULL

```

Hình 3.24. Ảnh minh họa tạo ràng buộc Not NULL.

Kiểm tra: Trong trường hợp này, bảng XUATXU được để tạo ràng buộc không bỏ trống. Nếu để trống, hệ thống sẽ báo lỗi.

```

49 INSERT INTO LOAICAY VALUES ('LC06','Cay hoa linh lan',NULL,'340000')
50
51

```

Data Output Explain Messages Notifications

ERROR: null value in column "xuatxu" of relation "loaicay" violates not-null constraint
DETAIL: Failing row contains (LC06, Cay hoa linh lan, null, \$340,000.00).
SQL state: 23502

Hình 3.25 Chạy thử minh họa chạy ràng buộc Not NULL.

2.2.VIEWS

- **Định nghĩa:** Là một bảng ảo trong cơ sở dữ liệu có nội dung được định nghĩa thông qua một câu lệnh SQL nào đó.

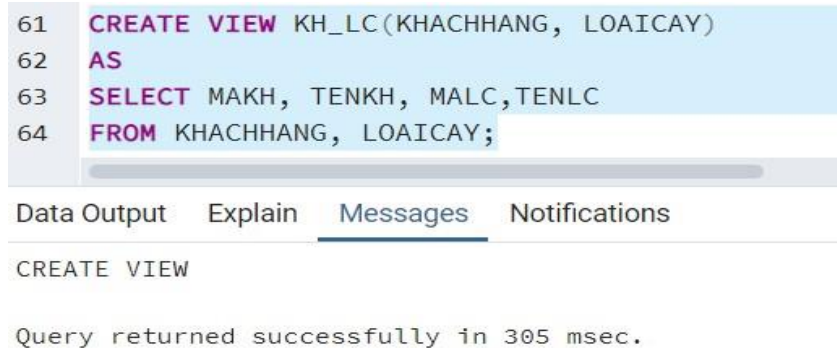
- Trong PostgreSQL có 4 view chính :

- Regular view (view thường) : cho phép chúng ta lưu trữ các truy vấn SQL phức tạp nhưng không cho phép bạn cập nhập dữ liệu đã được lưu. View thường thường được sử dụng để khắc phục những bất tiện của việc lặp lại các truy vấn phức tạp
- Materialized View (view vật liệu): được sử dụng để lưu trữ kết quả tính toán của một truy vấn, giống như bảng thực tế. Khi truy vấn được thực thi, kết quả được lưu trữ trong view này, do đó truy vấn tiếp theo trên view vật liệu sẽ nhanh hơn vì không cần phải tính toán lại các dữ liệu
- Updatable Views (view có thể cập nhập): cho phép người dùng cập nhập trực tiếp trên các bảng ghi trên view. Tuy nhiên, để updatable view làm việc, view phải được cấu trúc để cho phép trích xuất đầy đủ thông tin và có thể cập nhập lại dữ liệu
- Recursive Views (view đệ quy): là view cho phép thực hiện các việc liên kết đệ quy với chính chúng. Các view đệ quy trong PostgreSQL được sử dụng để xử lý các câu hỏi về phân cấp, nếu các quan hệ giữa các bản ghi có mối quan hệ đệ quy với chính chúng

2.2.1. REGULAR VIEW

- Chức năng: Tạo bảng VIEW.

Ví dụ: Câu lệnh trên bảng VIEW cho hai bảng KHACHHANG và LOAICAY bao gồm các cột MAKH, TENKH, MALC, TENLC.



Hình 3.26 Ảnh minh họa tạo bảng View.

Kiểm tra: Truy xuất bảng KH_LC.

66	SELECT * FROM KH_LC			
Data Output Explain Messages Notifications				
	khachhang character (4)	loaicay character varying (40)	malc character (4)	tenlc character varying (40)
1	KH01	Liz Kim Cuong	LC01	Xuong rong tai tho
2	KH02	Ivone Dieu	LC01	Xuong rong tai tho
3	KH03	Emma Nhat Khanh	LC01	Xuong rong tai tho
4	KH01	Liz Kim Cuong	LC02	Sen thach ngoc
5	KH02	Ivone Dieu	LC02	Sen thach ngoc
6	KH03	Emma Nhat Khanh	LC02	Sen thach ngoc
7	KH01	Liz Kim Cuong	LC03	Ba mau rau
8	KH02	Ivone Dieu	LC03	Ba mau rau

Hình 3.27 Truy xuất bảng View.

2.2.2. UPDATATABLE VIEW

- Chức năng: Kiểm tra tính toàn vẹn của dữ liệu khi INSERT và UPDATE dữ liệu vào.

Ví dụ: Câu lệnh trên kiểm tra cột DIACHI không được để giá trị NULL.



Hình 3.28 Kiểm tra View.

2.2.3. MATERIALIZED VIEW

- Chức năng: lưu trữ kết quả tính toán của một câu truy vấn

Ví dụ: Tính doanh thu của từng tháng theo từng rạp năm 2022

```
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

Hình 3.29 Materialized View.

2.2.4. RECURSIVE VIEW

- Chức năng: liên kết đệ quy với chính chúng

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

Hình 3.30. Recursive View.

2.2.5. DROP VIEW

- Chức năng: Để xóa bảng VIEW đã tạo.

Ví dụ: Xóa bảng view CHECK_KH tạo ở mục 3.2.2.

```
88 DROP VIEW CHECK_KH
```

Data Output Explain Messages Notifications

DROP VIEW

Query returned successfully in 95 msec.

Hình 3.31 Drop View.

2.3. INDEX

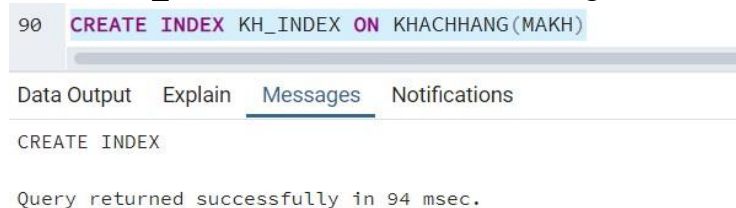
- **Định nghĩa:** Là một cấu trúc dữ liệu được dùng để định vị và truy cập nhanh nhất vào dữ liệu trong các bảng database, tăng nhanh thời gian hiệu suất và truy xuất dữ liệu.
- INDEX giúp tăng tốc các câu truy vấn SELECT chứa các mệnh đề WHERE hoặc ORDER, nhưng nó làm chậm việc dữ liệu nhập vào với các lệnh UPDATE và INSERT.
- Có 6 loại INDEX chính là :

- **B-tree Index** : B-tree là một cấu trúc dữ liệu cây cân bằng được sử dụng để lưu trữ các chỉ mục. Chỉ mục B-tree là loại chỉ mục phổ biến nhất trong PostgreSQL và được sử dụng để cải thiện hiệu suất truy vấn cho các truy vấn sử dụng toán tử so sánh như $=, <, >, <=, >=$
- **Hash Index**: Hash Index được sử dụng để cải thiện hiệu suất truy vấn cho các truy vấn sử dụng toán tử so sánh bằng ($=$). Chỉ mục Hash Index được tạo ra từ một hàm băm, cho phép tìm kiếm bản ghi nhanh hơn khi sử dụng phép so sánh bằng.
- **GiST Index** : là một chỉ mục đa chiều, được sử dụng cho các dạng dữ liệu không phải là số, chẳng hạn như đối tượng không gian hai hoặc ba chiều, mảng, văn bản hoặc các đối tượng khác cần được xử lý dưới dạng phân cấp
- **GIN Index** : là một loại chỉ mục đa chiều, sử dụng cho các cấu trúc dữ liệu tập hợp và mảng.
- **SP-GiST Index** : là một loại chỉ mục đa chiều, được sử dụng cho các cấu trúc dữ liệu phân tách không gian
- **BRIN Index** : được sử dụng để cải thiện hiệu suất truy vấn cho các bảng dài liên tục, nhưng không thích hợp cho các bảng với các giá trị ngẫu nhiên

2.3.1. CREATE INDEX

- Chức năng: Tạo Index.

Ví dụ: Tạo chỉ mục Index KH_INDEX trên cột MAKH của bảng KHACHHANG.



Hình 3.32 Create index.

```
CREATE INDEX KH_MAKH ON KHACHHANG USING BTREE(TEN);
```

Hình 3.33 Create B-Tree index.

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

Hình 3.34 Create Hash index.

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

Hình 3.35. Create GiST index.

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

Hình 3.36 Create GIN index.

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

Hình 3.37 Create SP-Gist index.

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

Hình 3.38 Create BRIN index.

Hình 3.39 Tạo giao diện.

2.3.2. DROP INDEX

- Chức năng: Xóa chỉ mục Index đã tạo.

Ví dụ: Xóa chỉ mục KH_INDEX trên cột MAKH của bảng KHACHHANG đã tạo ở mục 3.3.1.

Hình 3.40 Drop Index.

2.4. TRIGGER

- Chức năng: Là một thủ tục SQL được thực thi ở phía server khi có một sự kiện như Insert, Delete hay Update.

Ví dụ: Tạo ràng buộc TRIGGER kiểm tra giá trị nhập vào của bảng LOAICAY có đầy đủ cột hay không.

```
CREATE OR REPLACE FUNCTION CHECK_XUATXU_GIA_4()
RETURNS TRIGGER
LANGUAGE PLPGSQL
AS
$$
BEGIN
    INSERT INTO LOAICAY(MALC, TENLC, XUATXU, GIA) VALUES (new.MALC, new.TENLC, new.XUATXU, new.GIA);

    RETURN NEW;
END;
$$

INSERT INTO LOAICAY VALUES ('LC09','Sen thach ngoc','Anh','23000')
```

Output Explain Messages Notifications

RT 0 1

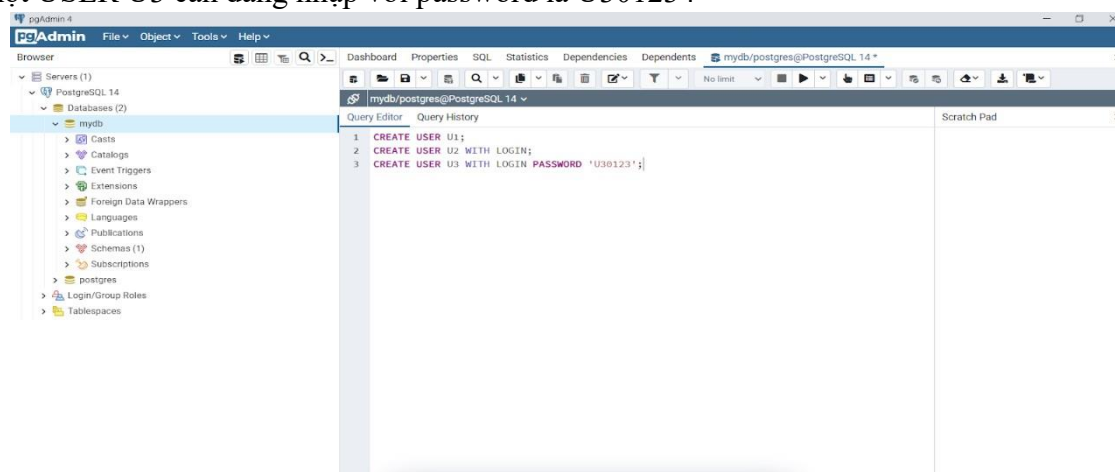
Hình 3.41 Trigger.

2.5.Chức thực và phân quyền

2.5.1. TẠO USER

- PostgreSQL đã tích hợp việc tạo USER và ROLE lại với nhau vì thế ta chỉ cần tạo ROLE thôi . PostgreSQL có hai loại ROLE đó là LOGIN ROLE và NON-LOGIN ROLE.
- Mặc định thì khi tạo một USER (hay ROLE) trong Postgresql thì thường là tạo một NON-LOGIN ROLE với câu lệnh: CREATE ROLE/USER username;
- Để tạo một LOGIN ROLE ta dùng câu lệnh: CREATE ROLE/USER username WITH LOGIN;
- Để cài đặt Password cho LOGIN ROLE ta dùng câu lệnh: CREATE ROLE/USER username WITH LOGIN PASSWORD 'Passworduser'; Trong đó:
username: tên tài khoản người dùng.
Passworduser: mật khẩu đặt cho tài khoản người dùng.

Ví dụ: trong hình ta đang tạo một USER U1 không cần đăng nhập, một USER U2 cần đăng nhập và một USER U3 cần đăng nhập với password là U301234



Hình 3.42 Ảnh minh họa tạo User.

Và để xóa USER (hay ROLE) trong PostgreSQL ta sử dụng câu lệnh: DROP ROLE username;

2.5.2. PHÂN QUYỀN

- GRANT
 - Cung cấp cho người dùng các quyền trên Database, Table, Schema: GRANT Privileges ON Object Object_name TO username; Trong đó:

Privileges: nhóm các đặc quyền.

Object: Database, Table, Schema.

Username: tên người dùng.

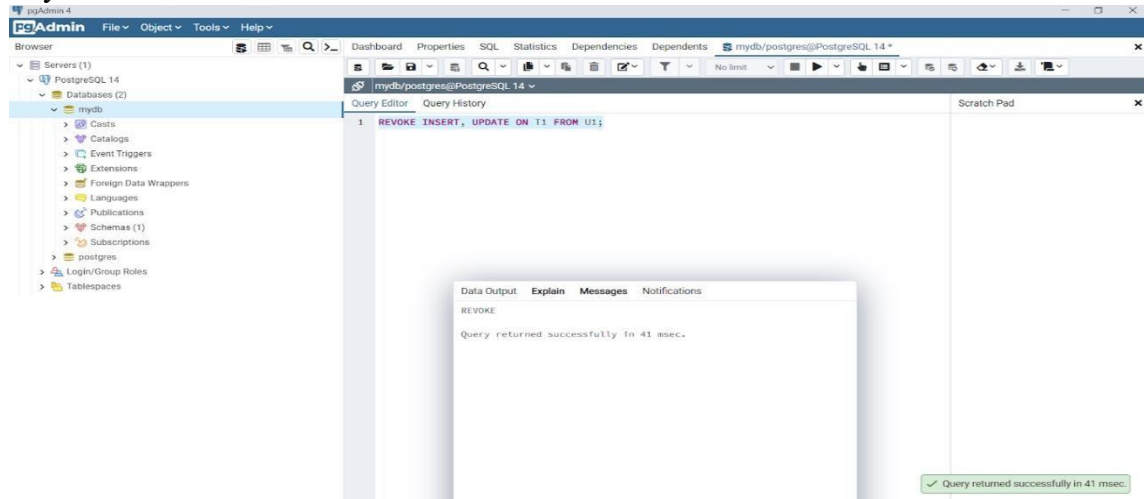
- Tạo người trạng thái siêu người dùng: ALTER USER username WITH SUPERUSER;
 - Xóa trạng thái siêu người dùng: ALTER USER username WITH NOSUPERUSER;
 - Cấp tạo một USER con từ USER cha: GRANT USERA TO USERB;
- REVOKE
REVOKE Privileges ON table FROM username; Trong đó:

Privileges: là nhóm các quyền gồm: SELECT, INSERT, UPDATE, DELETE, TRUNCATE, TRIGGER, CREATE, ALL.

Table: tên bảng.

username: tên người dùng.

Ví dụ: chúng ta sẽ thử xóa đặc quyền U1 Insert và Update trong bảng T1 như trong hình dưới đây:



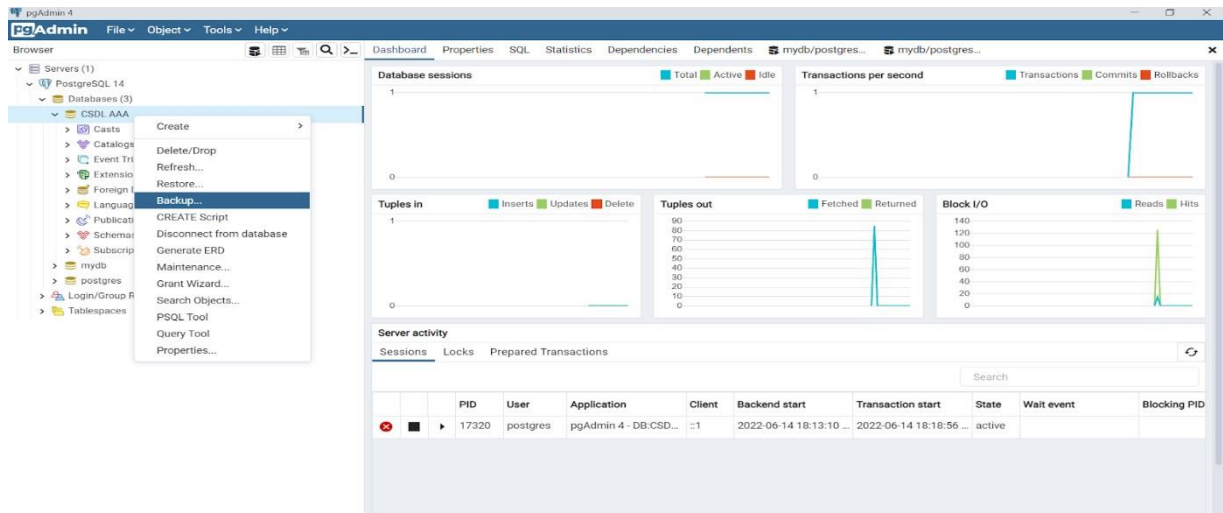
Hình 3.43 Ảnh minh họa tạo đặc quyền.

2.6 BACKUP VÀ RESTORE

2.6.1. Backup

Bước 1: Vào Database > Backup

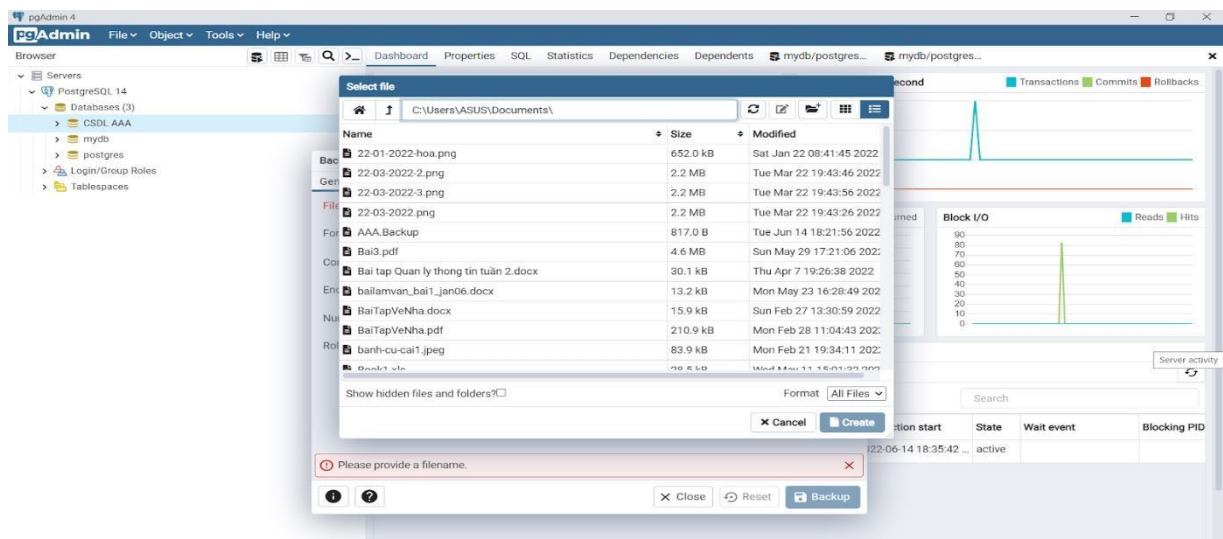
Ví dụ: Giả sử chúng ta có Database có tên là CSDL AAA. Chúng ta sẽ vào CSDL AAA để sao lưu thử.



Hình 3.44 Ảnh minh họa Backup.

Bước 2: Chọn đường dẫn lưu file

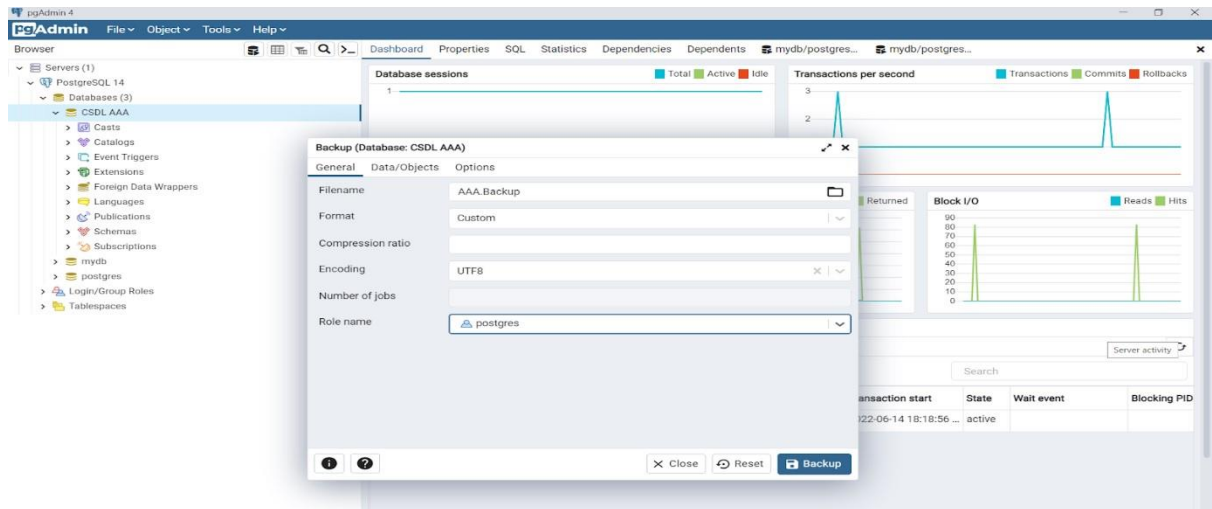
Ví dụ: chúng ta chọn đường dẫn là C:\USER\ASUS\Documents\



Hình 3.45 Ảnh minh họa chọn đường dẫn.

Bước 3: Do có nhiều kiểu định dạng (format) khi bạn sao lưu, chẳng hạn Custom, Tar, Directory, Plain.

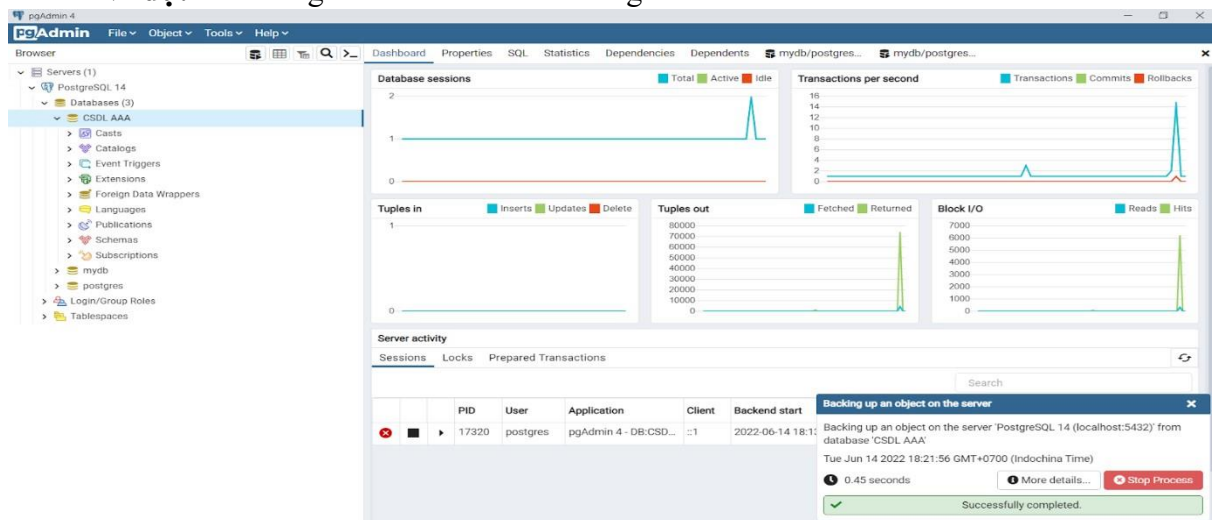
Ví dụ: Do định dạng Custom được ưa chuộng nhất, kết quả bạn sẽ có được một tập tin với phần mở rộng là backup. Chọn định dạng Custom, chọn vị trí, và tên tập tin sẽ được tạo ra.



Hình 3.46 Ảnh minh họa chọn định dạng kiểu, vị trí và tên cho tệp.

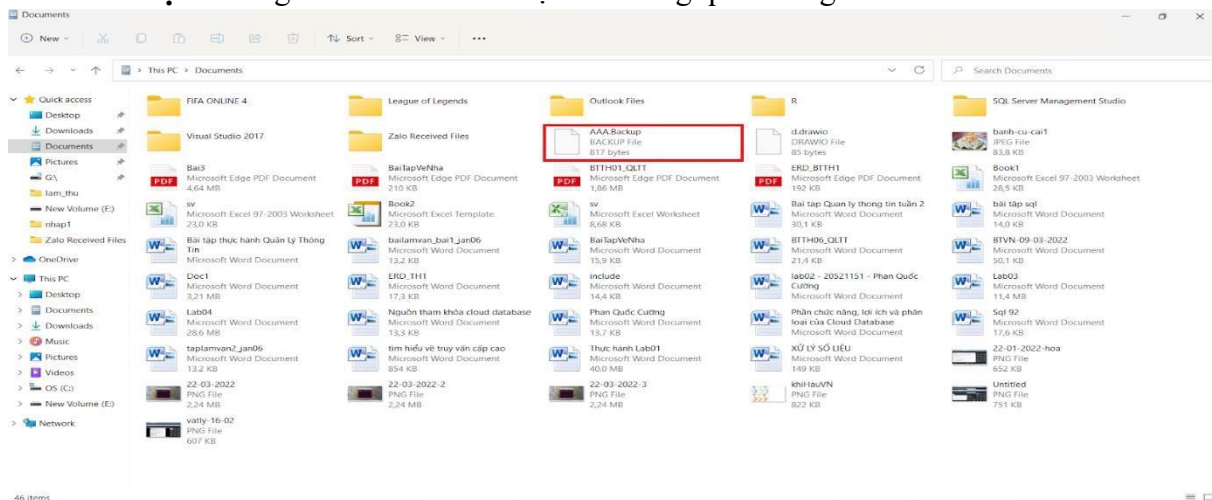
Bước 4: Khi sao lưu thành công sẽ hiện successfully

Ví dụ: khi chúng ta đã sao lưu thành công



Hình 3.47 Ảnh minh họa khi thành công.

Ví dụ : Chúng ta có thể kiểm tra lại file thông qua đường dẫn

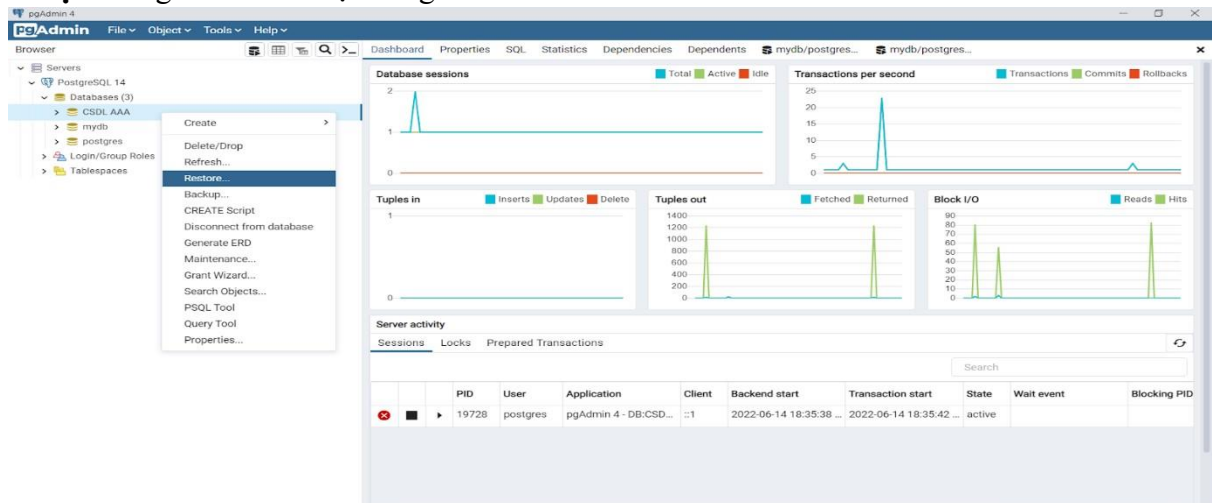


Hình 3.48 Ảnh minh họa file cần kiểm tra.

2.6.2. Restore

Bước 1: Vào Database chọn restore

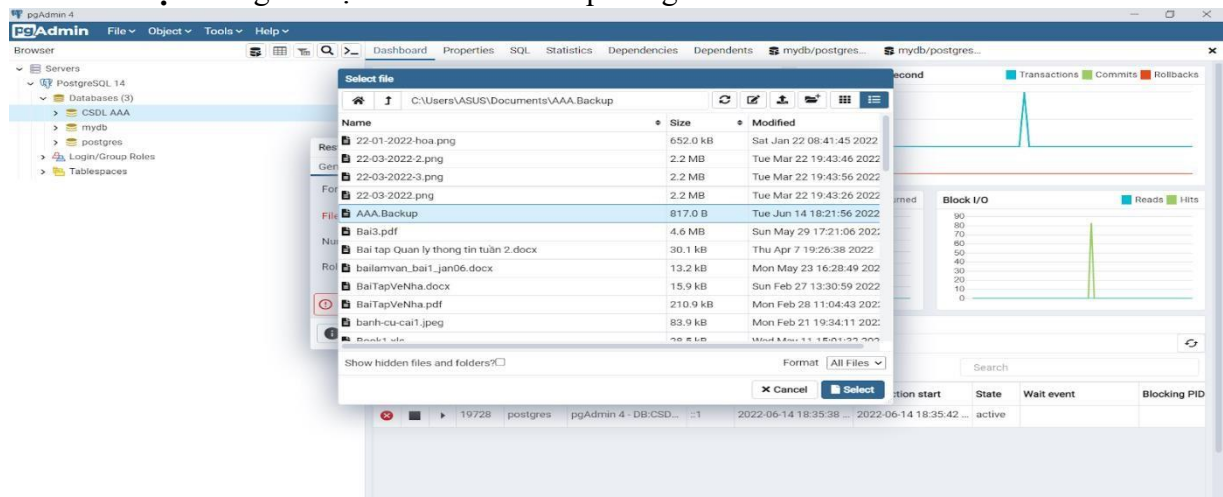
Ví dụ: chúng ta sẽ restore lại trong CSDL AAA



Hình 3.49 Ảnh minh họa chọn restore.

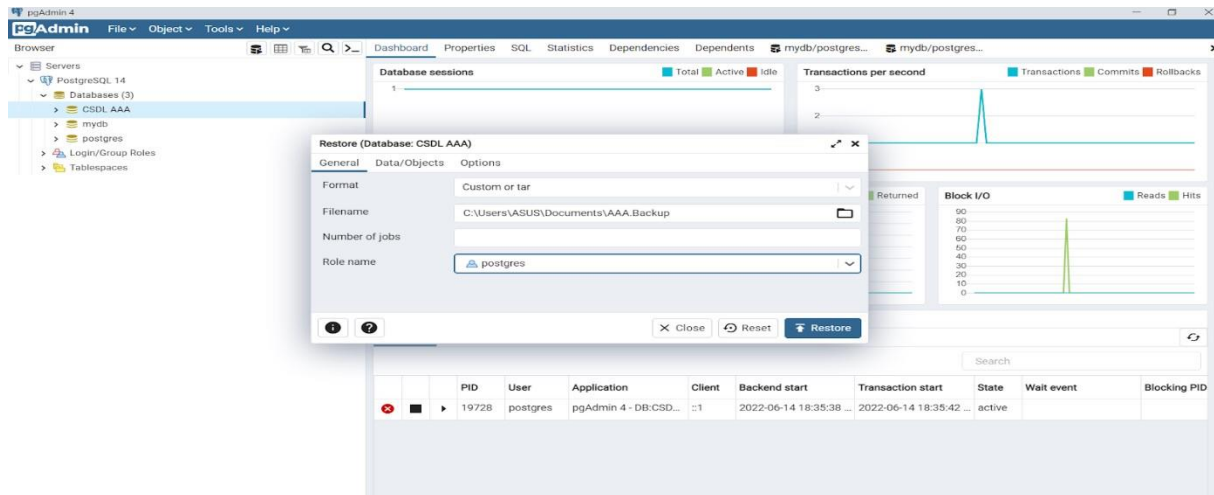
Bước 2: Chọn đường dẫn vào file đã sao lưu và chọn file

Ví dụ: chúng ta chọn file AAA.Backup trong C:\Users\ASUS\Documents



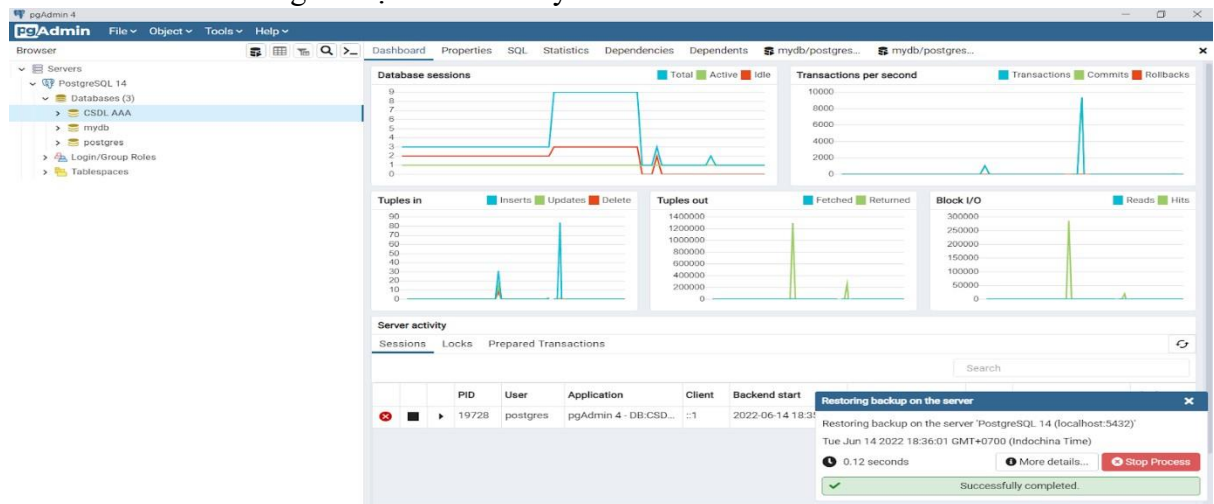
Hình 3.50 Ảnh minh họa chọn file trong đường dẫn.

Bước 3: Chọn restore để phục hồi dữ liệu



Hình 3.51 Ảnh minh họa restore.

Bước 4: Khi thành công sẽ hiện successfully

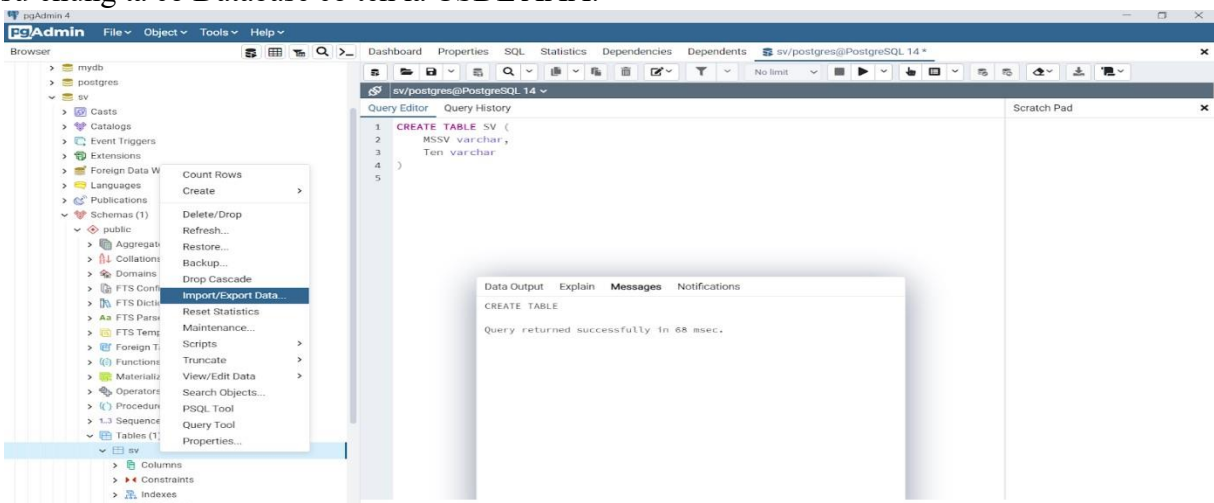


Hình 3.52 Ảnh minh họa restore thành công.

2.7.IMPORT VÀ EXPORT

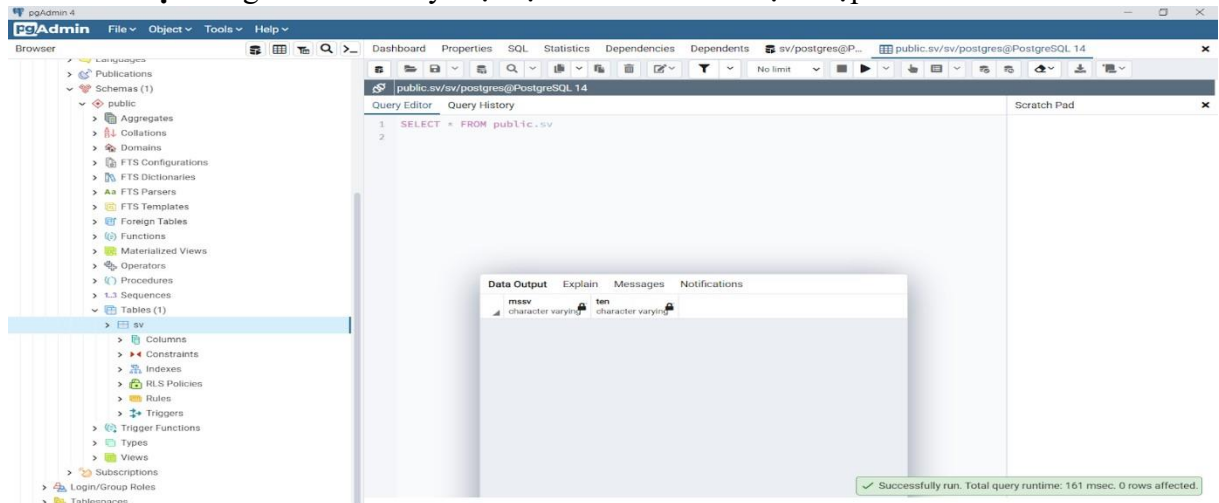
2.7.1. Import

Bước 1: Database > Schemas > Tables > chọn table > Import/Export Database Giả sử chúng ta có Database có tên là CSDL AAA.



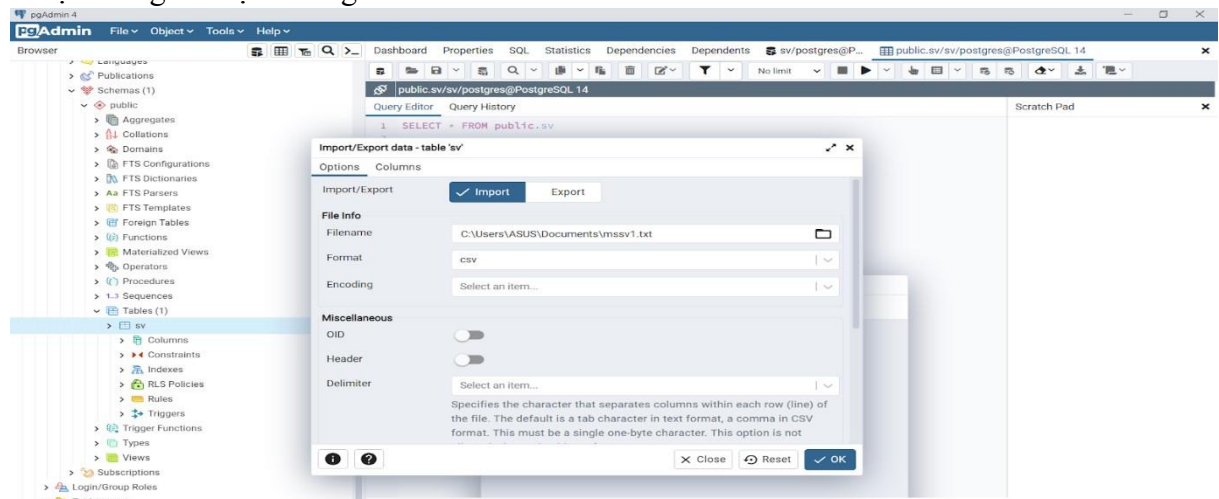
Hình 3.53 Ảnh minh họa chọn Import Database.

Ví dụ: trong hình dưới đây hiện tại SV chưa có số liệu nhập vào



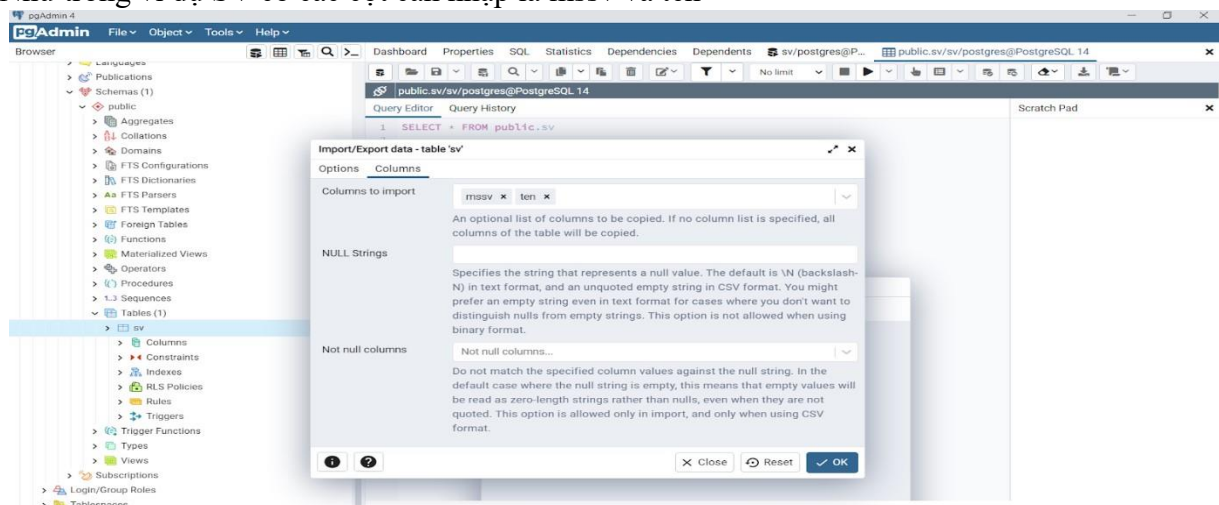
Hình 3.54 Ảnh minh họa chưa có số liệu trong bảng.

Bước 2: Chọn Import rồi chọn đường dẫn vào file chứa thông tin cần nhập và chọn
Ví dụ: chúng ta chọn đường dẫn C:\Users\ASUS\Documents\mssv1.txt



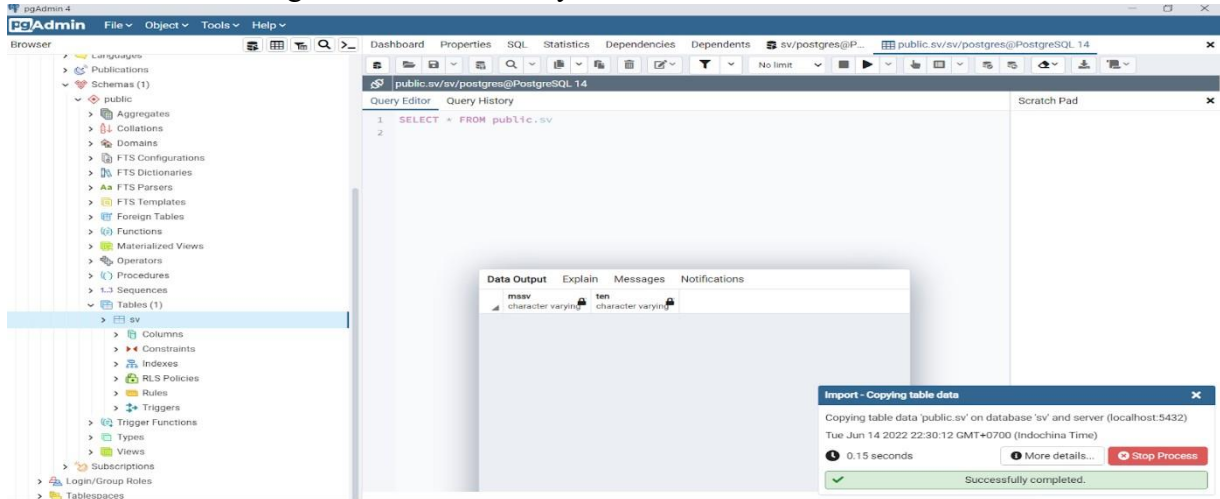
Hình 3.55 Ảnh minh họa chọn file và đường dẫn

Bước 3: Chọn những cột cần nhập thông tin
Như trong ví dụ SV có các cột cần nhập là mssv và ten



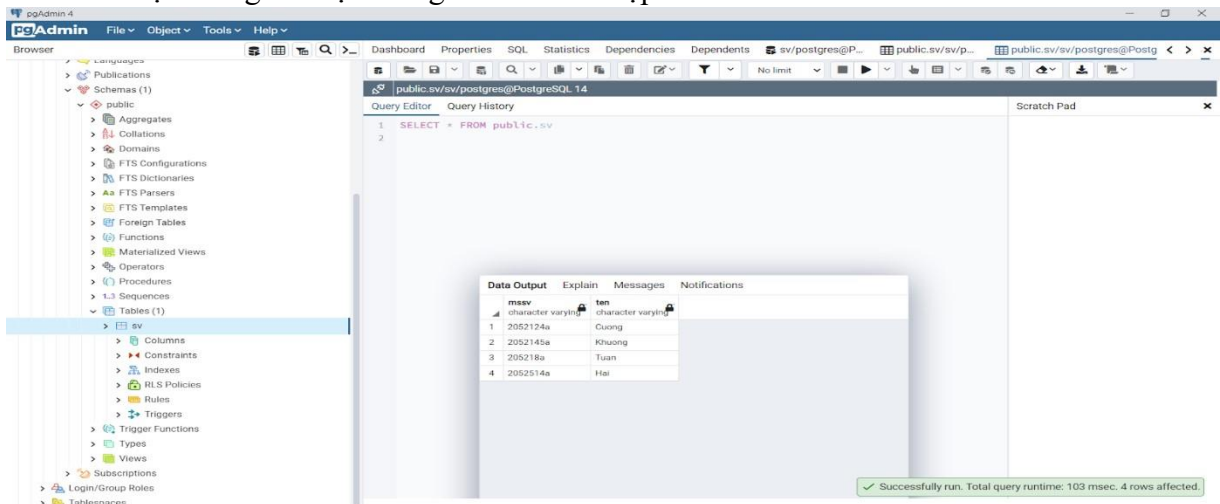
Hình 3.56 Ảnh minh họa chọn thông tin cần nhập

Bước 4: Khi thành công hiện thị successfully



Hình 3.57. Ảnh minh họa Import thành công

Ví dụ: chúng ta được thông tin sau khi nhập

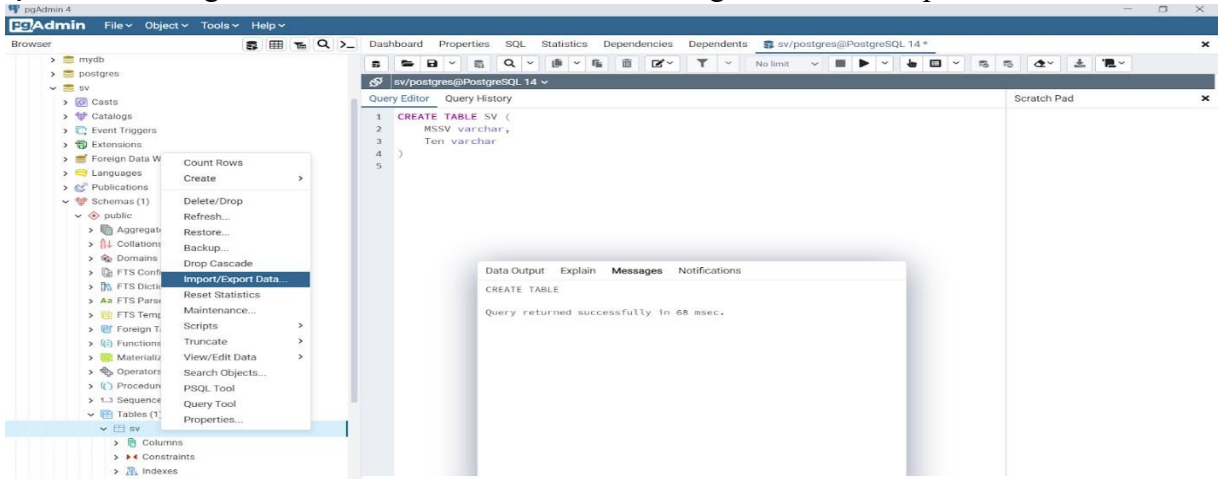


Hình 3.58 Ảnh minh họa Import thành công

2.7.2. Export

Bước 1: Database > Schemas > Tables > chọn table > Import/Export Database

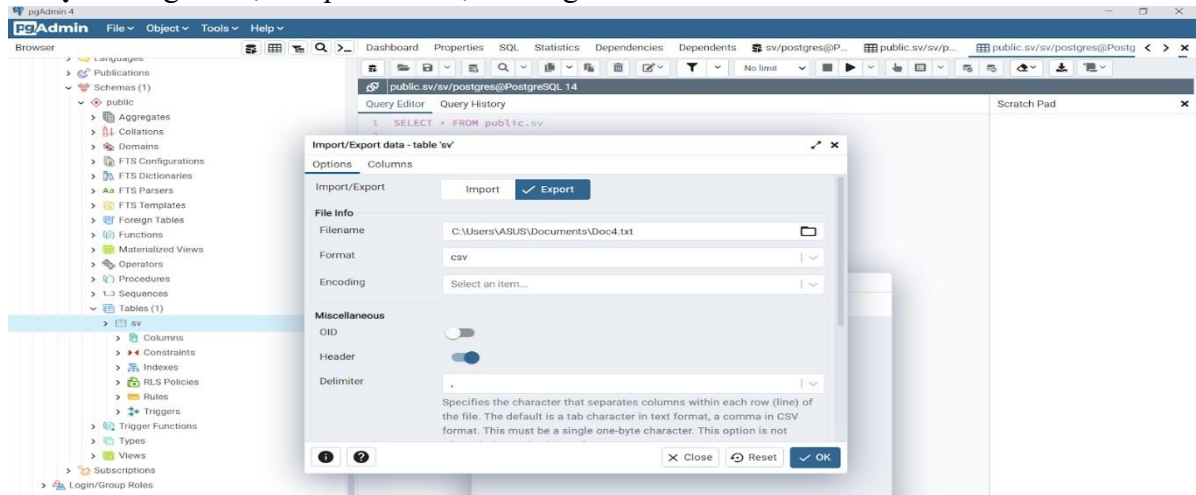
Ví dụ: Giả sử chúng ta có Database có tên là SV và có bảng là SV, chọn Export.



Hình 3.59 Ảnh minh họa chọn Export Database.

Bước 2: Chọn Export rồi chọn đường dẫn vào file cần xuất thông tin và chọn file, cuối cùng nhấn ok

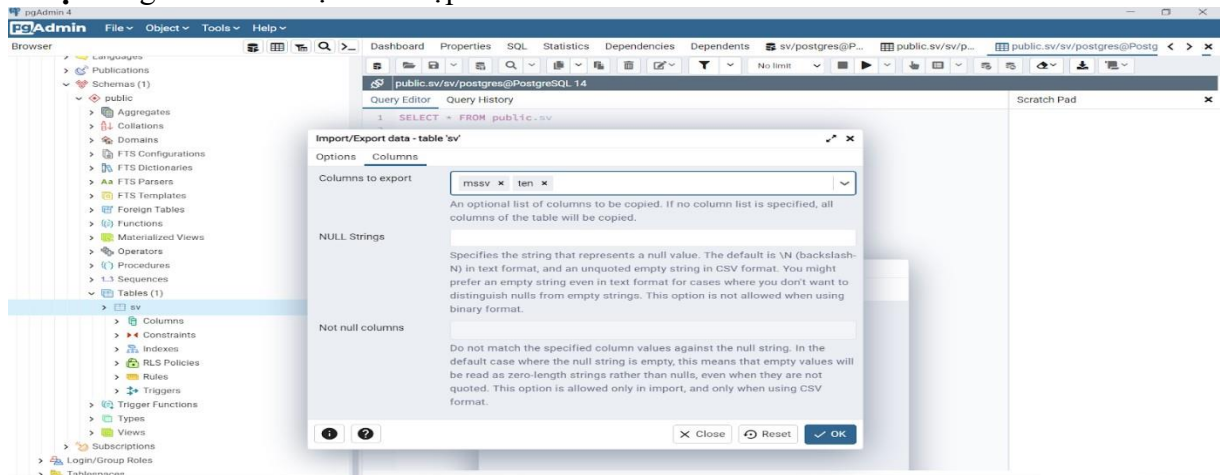
Ví dụ: chúng ta chọn Export rồi chọn đường dẫn C:\Users\ASUS\Documents\Doc4.txt



Hình 3.60 Ảnh minh họa chọn file trong đường dẫn.

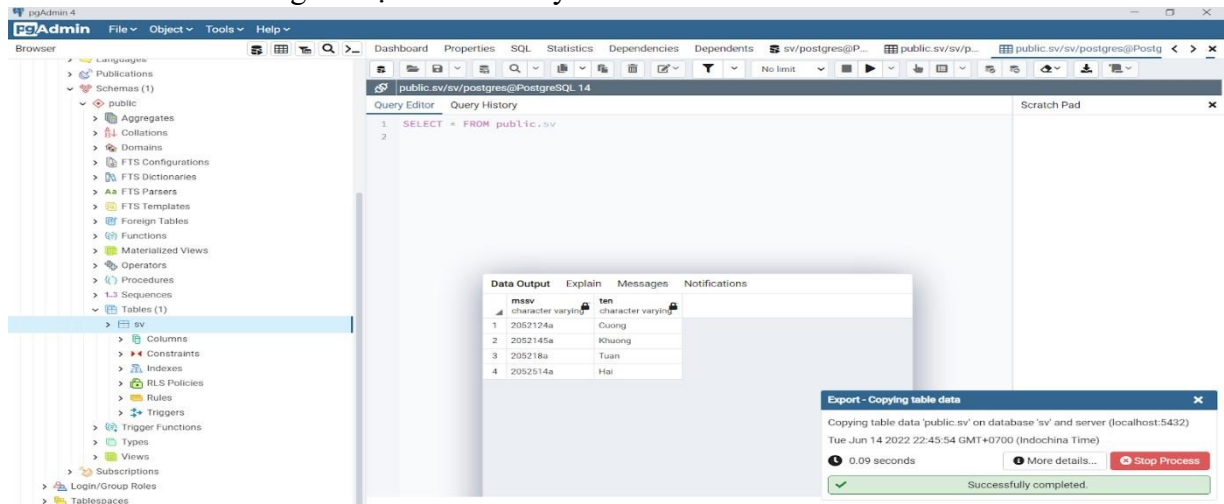
Bước 3: Chọn những cột cần xuất thông tin

Ví dụ: bảng SV có các cột cần nhập là mssv và ten



Hình 3.61 Ảnh minh họa chọn thông tin cần xuất.

Bước 4: Khi thành công sẽ hiện successfully



Hình 3.62 Ảnh minh họa xuất thành công

Hình dưới đây là Doc4 sau khi xuất thông tin từ bảng SV



Hình 3.63 Ảnh minh họa thông tin được xuất vào file Doc4

2.8 Stored Procedure

- **Định nghĩa:** là bao gồm các câu lệnh Transact-SQL và được lưu lại trong cơ sở dữ liệu. Để thực thi lệnh chỉ cần gọi ra. Transact-SQL (T-SQL) là một ngôn ngữ lập trình được sử dụng là trung gian giữa cơ sở dữ liệu và các ứng dụng. Nó tương đối dễ học vì thực chất nó được tạo bởi hầu hết là các lệnh SQL
- **Lợi ích :** Module hóa, thực thi nhanh hơn, giảm tải băng thông
- **Cú pháp :**

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

- **Gọi hàm :**
`call procedure_name(argument_list);`

Hình 3.64 Ảnh minh họa cú pháp Stored Procedure

- **Ví dụ :** Viết procedure xử lý việc chuyển tiền, trừ số tiền có trong người gửi và cộng tiền cho người nhận

```

create or replace procedure transfer(
    sender int,
    receiver int,
    amount dec
)
language plpgsql
as $$
begin
    -- trừ số tiền có trong người gửi
    update accounts
    set balance = balance - amount
    where id = sender;

    -- cộng số tiền có trong người nhận
    update accounts
    set balance = balance + amount
    where id = receiver;

    commit;
end;$$

```

Hình 3.65 Ảnh minh họa ví dụ Store Procedure

- Gọi store procedure

```
call transfer(1,2,1000);
```

Hình 3.66 Hàm gọi ví dụ Store Procedure

- Kết quả trước khi thực hiện procedure :

	id integer	name character varying (100)	balance numeric (15,2)
1	1	Bob	10000.00
2	2	Alice	10000.00

Hình 3.67 Bảng trước khi thực hiện procedure

- Kết quả sau khi thực hiện procedure

	id [PK] integer	name character varying	balance numeric (15,2)
1	1	Bob	9000.00
2	2	Alice	11000.00

Hình 3.68 Bảng sau khi thực hiện procedure

2.9 FUNCTION

- **Định nghĩa:** PostgreSQL Function (còn được gọi là Store Procedures) được sử dụng để thực thi các câu lệnh SQL để thực hiện một mục đích nhất định. Nó cho phép tái sử dụng bằng cách gọi function thay vì phải viết lại câu SQL.

- **Lợi ích :** Module hóa, thực thi nhanh hơn

- **Cú pháp :**

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

Hình 3.69 Cú pháp sử dụng FUNCTION

- **Gọi hàm :**

```
SELECT function_name();
```

Hình 3.70 Hàm gọi ví dụ gọi FUNCTION

- **Xóa hàm:**

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

Hình 3.71 Hàm xóa FUNCTION

-Ví dụ: Đếm số lượng nhân viên có trong bảng nhân viên

staff_id integer	fullname character varying	position character varying	description text	created_at timestamp without time zone	updated_at timestamp with time zone	deleted_at timestamp with time zone
1	Nguyễn Văn A	admin	Quản trị viên vinasupport.com	2019-09-29 14:30:10.641892	[null]	[null]
2	Nguyễn Văn B	admin	Quản trị viên vinasupport.com	2019-09-29 14:35:59.333119	[null]	[null]
3	Nguyễn Văn C	developer	Lập trình website vinasupport.com	2019-09-29 14:35:59.333119	[null]	[null]
4	Nguyễn Thị D	design	Thiết kế website vinasupport.com	2019-09-29 14:35:59.333119	[null]	[null]
5	Nguyễn Văn E	author	Quản trị nội dung website vinasupport.com	2019-09-29 14:35:59.333119	[null]	[null]
6	Nguyễn Văn F	author	Quản trị nội dung website vinasupport.com	2019-09-29 14:35:59.333119	[null]	[null]
7	Nguyễn Văn G	author	Quản trị nội dung website vinasupport.com	2019-09-29 14:35:59.333119	[null]	[null]
8	Nguyễn Văn H	author	Quản trị nội dung website vinasupport.com	2019-09-29 14:35:59.333119	[null]	[null]
9	Nguyễn Thị I	collaborator	Cộng tác viên vinasupport.com	2019-09-29 14:35:59.333119	[null]	[null]
10	Nguyễn Thị J	collaborator	Cộng tác viên vinasupport.com	2019-09-29 14:35:59.333119	[null]	[null]
11	Nguyễn Thị Q	collaborator	Cộng tác viên vinasupport.com	2019-09-29 14:35:59.333119	[null]	[null]
12	Nguyễn Thị K	collaborator	Cộng tác viên vinasupport.com	2019-09-29 14:35:59.333119	[null]	[null]

Hình 3.72 Dữ liệu được có trong bảng nhân viên

- Hàm function

```

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();

```

Hình 3.73 Hàm FUNCTION đếm số lượng nhân viên

- Kết quả

Data Output	Explain	Messages	Notifications
getstaffcount integer			
1	12		

Hình 3.74 Kết quả số lượng nhân viên trả về

CHƯƠNG 4 : CSDL QUẢN LÝ RẠP CHIẾU PHIM

1. Mô tả CSDL

- Cơ sở dữ liệu quản lý các rạp chiếu phim của một công ty. Mỗi rạp trong cơ sở dữ liệu sẽ bao gồm Mã rạp, Tên Rạp, Địa chỉ, Điện thoại, số phòng, tổng số ghế
- Mỗi rạp sẽ có nhiều phòng chiếu. Mỗi phòng chiếu sẽ thuộc một rạp. Mỗi phòng chiếu sẽ có Mã phòng, Tên phòng và tổng số ghế
- Cơ sở dữ liệu cũng quản lý các bộ phim, các bộ phim được chiếu trên toàn bộ hệ thống của rạp. Mỗi bộ phim sẽ gồm Tên Phim, Mã nước sản xuất, Mã hãng sản xuất, Đạo diễn, Mã thể loại, Ngày khởi chiếu, Ngày kết thúc, Tổng thu
- Các thông tin của phim về Hãng sản xuất, Nước sản xuất, Thể loại đều có bảng riêng gồm Mã và Tên
- Mỗi rạp cũng có lịch chiếu riêng và lịch sẽ có mã show. Mỗi mã show sẽ là một bộ phim được chiếu tại rạp đó với mã phim, mã phòng chiếu, mã rạp và mã giờ chiếu theo từng bộ với giờ chiếu khác nhau trong bảng Giờ chiếu
- Chi tiết cơ sở dữ liệu :
 - **Phim**(MaPhim, TenPhim, MaNuocSX, MaHangSX, DaoDien, 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)
 - **Ve** (MaVe, MaRap, MaPhong, MaGhe, MaShow, TrangThai)
 - **LichChieu** (MaShow, MaPhim, MaRap, MaPhong, NgayChieu, MaGioChieu, GiaVe, SoVeDaBan, TongTien)

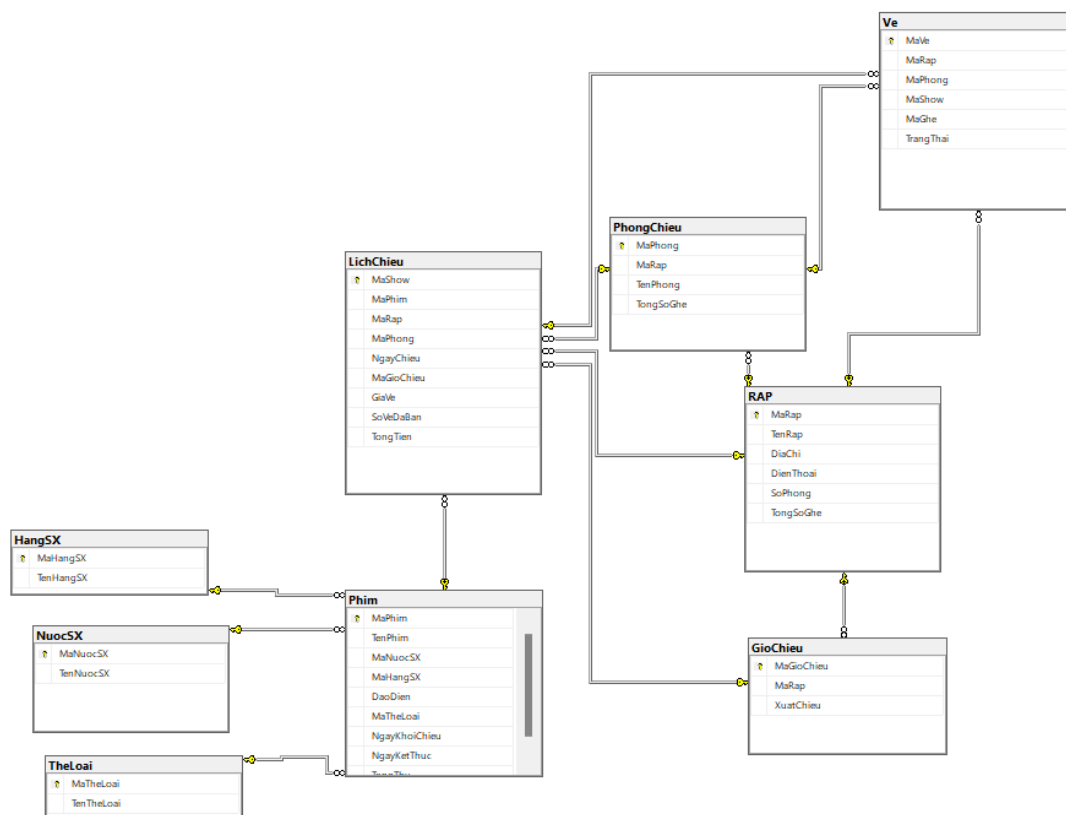
2. Thuộc tính của các thành phần

ĐỐI TƯỢNG	THÀNH PHẦN	THUỘC TÍNH
Phim	<u>MaPhim</u>	VARCHAR(10)
	TenPhim	VARCHAR(50)
	MaNuocSX	VARCHAR(10)
	MaHangSX	VARCHAR(10)
	DaoDien	VARCHAR(20)
	MaTheLoai	VARCHAR(10)
	NgayKhoiChieu	SMALLDATETIME
	NgayKetThuc	SMALLDATETIME
	TongThu	MONEY
HangSX	<u>MaHangSX</u>	VARCHAR(10)
	TenHangSX	VARCHAR(20)
NuocSX	<u>MaNuocSX</u>	VARCHAR(10)
	TenNuocSX	VARCHAR(20)
TheLoai	<u>MaTheLoai</u>	VARCHAR(10)
	TenTheLoai	VARCHAR(20)
RAP	<u>MaRap</u>	VARCHAR(10)
	TenRap	VARCHAR(50)
	DiaChi	VARCHAR(100)
	DienThoai	VARCHAR(10)

	SoPhong	VARCHAR(10)
	TongSoGhe	INT
GioChieu	<u>MaGioChieu</u>	VARCHAR(10)
	MaRap	VARCHAR(10)
	XuatChieu	TEXT
PhongChieu	<u>MaPhong</u>	VARCHAR(10)
	MaRap	VARCHAR(10)
	TenPhong	NVARCHAR(50)
	TongSoGhe	INT
Ve	<u>MaVe</u>	VARCHAR(10)
	MaRap	VARCHAR(10)
	MaPhong	VARCHAR(10)
	MaGhe	VARCHAR(10)
	MaShow	VARCHAR(10)
	TrangThai	VARCHAR(10)
LichChieu	<u>MaShow</u>	VARCHAR(10)
	MaPhim	VARCHAR(10)
	MaRap	VARCHAR(10)
	MaPhong	VARCHAR(10)
	NgayChieu	SMALLDATETIME
	MaGioChieu	VARCHAR(10)

	GiaVe	MONEY
	SoVeDaBan	INT
	TongTien	MONEY

3. Sơ đồ liên kết quan hệ của các bảng



Hình 4.1 Bảng quan hệ giữa các bảng

4. Video quay demo CSDL quản lý rạp chiếu phim

- link:

<https://drive.google.com/drive/folders/1ECLim0FSyGLftyxGBPSJyberUFAq7vP?usp=sharing>

CHƯƠNG 5 : ĐÁNH GIÁ VÀ KẾT LUẬN

1. Ưu và nhược điểm

a. Ưu điểm

- Tính bảo mật cao: PostgreSQL có giải pháp bảo mật được tích hợp sẵn trong hệ thống, cung cấp các tính năng bảo mật như truy cập kiểm soát, mã hóa và phân quyền dữ liệu.
- Truy vấn mạnh mẽ : PostgreSQL có khả năng hiệu quả trong việc xử lý các truy vấn phức tạp, hỗ trợ khả năng truy vấn song song, tra cứu văn bản và truy xuất đối tượng địa lý
- Khả năng mở rộng tốt : PostgreSQL hỗ trợ nhiều cơ chế mở rộng, bao gồm chia sẻ bảng, phân vùng và replica, cho phép nâng cao tính sẵn sàng và khả năng mở rộng hệ thống
- Linh hoạt : PostgreSQL có khả năng hỗ trợ nhiều loại dữ liệu như JSON, XML, và nhiều loại hỗ trợ ngôn ngữ lập trình khác như Java, Python, C++, Ruby và Perl
- Có khả năng chạy trang web, ứng dụng di động
- Lưu lại nhật ký, hình thành cơ sở dữ liệu hỗ trợ sửa lỗi.

b. Nhược điểm

- Khó sử dụng : PostgreSQL có cấu trúc phức tạp và khó học đối với người mới bắt đầu
- Hiệu suất chậm : PostgreSQL thường có hiệu suất chậm hơn so với một số hệ điều hành quản trị cơ sở dữ liệu khác trong một số trường hợp đặt biệt
- Khó khắc phục sự cố : Khi xảy ra lỗi, việc khắc phục vấn đề trong PostgreSQL thường phức tạp hơn nhiều so với một số hệ quản trị cơ sở dữ liệu khác

2. So sánh PostgreSQL và SQL Server

<u>Thành phần</u>	<u>PostgreSQL</u>	<u>SQL Server</u>
Mô hình	Mã nguồn mở và miễn phí, được duy trì bởi PostgreSQL Global Development Group	Phục vụ cho thương mại, được xây dựng và bảo trì bởi Microsoft
Hệ điều hành hỗ trợ	FreeBSD, HP-UX, Linux, NetBSD, OpenBSD, OS X, Solaris, Unix, Windows	Tương tự như PostgreSQL

Thủ tục lưu trữ	Sử dụng bằng ngôn ngữ đọc quyền PL / pgSQL hoặc với các ngôn ngữ phổ biến như Perl, Python, Tcl ...	Transact SQL và .NET
Phương pháp phân vùng	Phân vùng khai báo (theo phạm vi hoặc danh sách)	Phân phối trên một số tệp (Phần vùn ngang), bảo vệ theo liên đoàn
Khả năng viết kịch bản	Có thể điều khiển hoàn toàn từ dòng lệnh	Được điều khiển nhiều hơn thông qua GUI
Tình trạng cập nhật	Một năm vài lần	Thường xuyên cập nhật phiên bản mới

3. Nhận xét :

PostgreSQL là một hệ quản trị cơ sở dữ liệu mạnh mẽ và linh hoạt, được phát triển dựa trên nguyên tắc ACID (Atomicity, Consistency, Isolation, Durability) để đảm bảo tính toàn vẹn và đồng nhất của dữ liệu. Với khả năng xử lý hiệu suất cao, nó hỗ trợ đa luồng, có khả năng mở rộng và phù hợp với các ứng dụng từ quy mô nhỏ đến lớn.

Các loại hình công ty thường dùng PostgreSQL là đa dạng và phổ biến trong nhiều ngành công nghiệp như :Công ty phần mềm và công nghệ, công ty tài chính, ty thương mại điện tử, ty truyền thông và giải trí,...

Về dữ liệu, PostgreSQL có thể được sử dụng để lưu trữ và quản lý các loại dữ liệu khác nhau, bao gồm:

- Dữ liệu người dùng và thông tin cá nhân: PostgreSQL có thể lưu trữ thông tin khách hàng, tài khoản người dùng, lịch sử mua hàng và thông tin cá nhân khác một cách an toàn.
- Dữ liệu giao dịch: PostgreSQL hỗ trợ lưu trữ dữ liệu giao dịch, bao gồm đơn hàng, hóa đơn, thanh toán và chi tiết giao dịch.
- Dữ liệu địa lý và không gian: PostgreSQL có hỗ trợ đặc lực cho các loại dữ liệu địa lý và không gian, cho phép lưu trữ và truy vấn các thông tin vị trí, bản đồ và dữ liệu địa lý liên quan.
- Dữ liệu phương tiện: PostgreSQL có thể lưu trữ dữ liệu phương tiện như hình ảnh, video, âm thanh và các tệp tin đa phương tiện khác.
- Dữ liệu phân tích: PostgreSQL có thể được sử dụng để lưu trữ và phân tích dữ liệu lớn, bao gồm dữ liệu từ các hệ thống Business Intelligence (BI), dữ liệu từ công cụ phân tích dữ liệu và dữ liệu thu thập từ các nguồn khác nhau để tạo ra báo cáo và thông tin quản lý.

4. Kết luận

- Dựa trên bài báo cáo, ta có thể kết luận rằng PostgreSQL là một hệ quản trị cơ sở dữ liệu mã nguồn mở có các tính năng nâng cao và độ tin cậy cao. Nó có thể xử lý các tác vụ cơ bản và phức tạp một cách hiệu quả bằng các tính năng như đa phiên bản, tối ưu hóa truy vấn và phân phối cơ sở dữ liệu. Nó cũng hỗ trợ nhiều ngôn ngữ lập trình và có khả năng tích hợp với các ứng dụng và công nghệ khác. PostgreSQL cũng có cộng đồng lớn và năng động giúp đảm bảo rằng nó được phát triển và cập nhật liên tục, đáp ứng như cầu của người dùng ngày càng cao. Tóm lại, PostgreSQL là một lựa chọn tuyệt vời cho các dự án quản trị cơ sở dữ liệu lớn và phức tạp

TÀI LIỆU THAM KHẢO

- [1] B. Cloud, "PostgreSQL là gì? Tìm hiểu thông tin về cơ sở dữ liệu mã nguồn mở," 20 9 2018. [Online].
Available: <https://bizflycloud.vn/tin-tuc/postgresql-la-gi-tim-hieu-ve-co-so-du-lieu-ma-nguon-mo-tien-tien-nhat-the-gioi-20180919175924611.htm>.
- [2] ICT, "Hướng dẫn sử dụng, cài đặt và cấu hình Database PostgreSQL," 9 4 2019. [Online].
Available: <https://ictsharing.com/lap-trinh/huong-dan-su-dung-cai-at-va-cau-hinh.html>.
- [3] B. Hiếu, "Tạo user, DB và phân quyền cho PostgreSQL, tất cả những gì bạn cần biết," 20 9 2021. [Online].
Available: <https://viblo.asia/p/tao-user-db-va-phan-quyen-cho-postgresql-tat-ca-nhung-gi-ban-can-biet-naQZRk8jlvx>.
- [4] T. V. Linh, "Giới thiệu các câu truy vấn trong PostgreSQL qua các ví dụ," 08 10 2021. [Online].
Available: <https://techmaster.vn/posts/36773/gioi-thieu-cac-cau-truy-van-trong-postgresql-qua-cac-vi-du>.
- [5] vinasupport.com, "[PostgreSQL] Lệnh SELECT – Lệnh lấy dữ liệu (SELECT Query)," 29 09 2019. [Online].
Available: <https://vinasupport.com/postgresql-lenh-select-lenh-lay-du-lieu-select-query/>.
- [6] SAKURAI, "Tạo xóa và phân quyền user trong PostgreSQL," 18 08 2020. [Online].
Available: <https://fixloinhanh.com/tao-va-phan-quyen-user-trong-postgresql/>.

[7] Vietnix, "PostgreSQL là gì?," 20 07 2022. [Online].

Available: <https://vietnix.vn/postgresql-la-gi>.