TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

----- 🙡 🕮 🙣 -----



**BÁO CÁO BÀI TẬP LỚN**

***Đề tài:* Thiết kế cơ sở dữ liệu về dịch vụ vận hành xe bus**

Giảng viên: **Thầy Trần Văn Đăng**

Nhóm sinh viên thực hiện:

|  |  |  |
| --- | --- | --- |
| STT | Họ và tên | MSSV |
| 1 | Hà Thành Long | 20225875 |
| 2 | Bùi Thành Long | 20225874 |
| 3 | Bùi Tuấn Phong | 20225900 |

**Hà Nội, năm 2024**

**I.PHẦN I : Kịch bản thế giới thực**

**1.Ứng dụng của hệ CSDL :**

Quản lí và vận hành hệ thống xe bus

Cung cấp thông tin dịch vụ xe bus cho người dùng

Quản lí danh sách nhân viên trong công ty

**2.Các yêu cầu về dữ liệu cần lưu trữ :**

Dữ liệu về các nhân viên trong công ty , gồm các thông tin về : mã nhân viên , họ và tên , chức vụ , số điện thoại , …

Dữ liệu về các ca làm việc của nhân viên

Dữ liệu về lộ trình các tuyến của xe bus

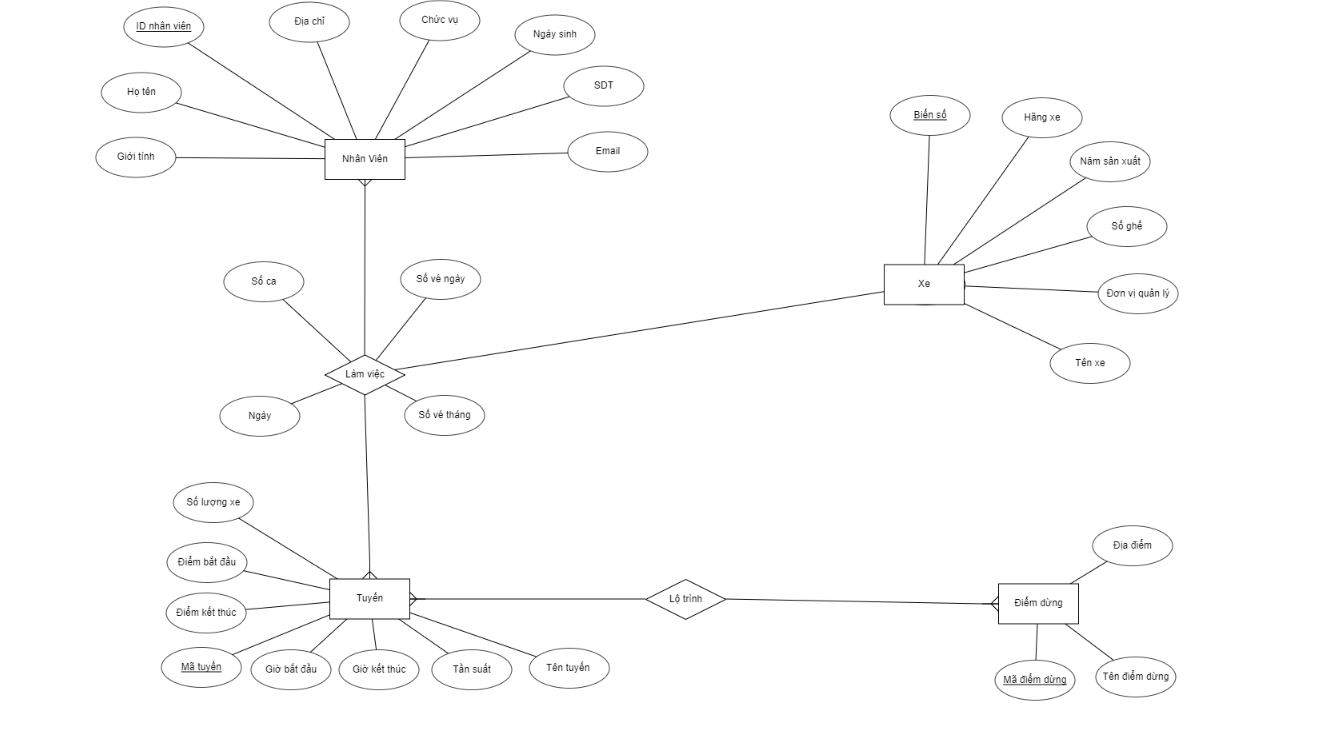
Dữ liệu về các tuyến vận hành của xe bus

Dữ liệu và thông tin về xe bus , bao gồm : biển số , hãng xe , năm sản xuất , số ghế , đơn vị quản lý , …

Dữ liệu về các điểm dừng xe bus trong thành phố , bao gồm : mã điểm dừng , tên điểm dừng , địa chỉ , …

II. PHẦN II : Lược đồ E-R

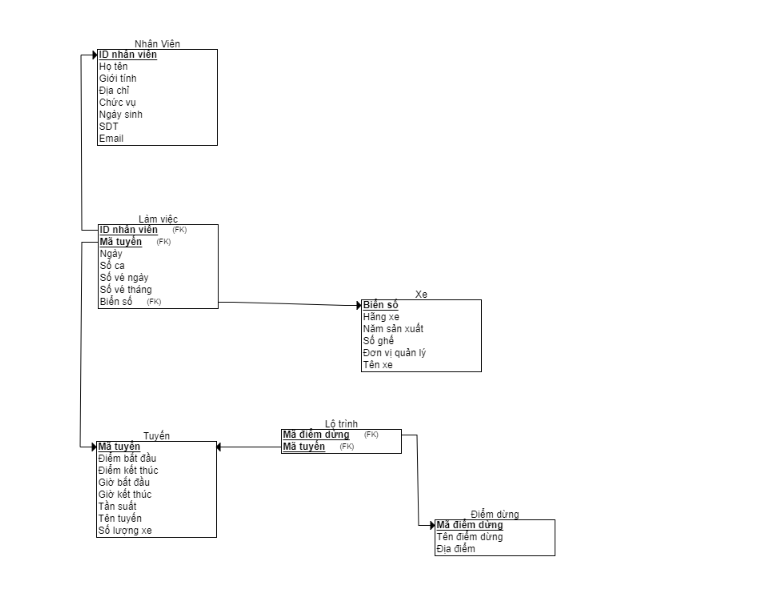
1. Chi tiết các bảng:



Lược đồ E-R gồm 4 tập thực thể : trong đó có 2 tập thực thể kết hợp .

III. PHẦN III : Chuyển lược đồ E-R sang các lược đồ quan hệ

Từ lược đồ E-R gồm 4 tập thực thể , … thuộc tính đa trị , chuyển thành 6 lược đồ quan hệ :



IV.PHẦN IV : Chuẩn hóa các lược đồ quan hệ về dạng chuẩn 3NF

Từ 6 lược đồ quan hệ ta chuẩn hóa về dạng chuẩn 3 NF .

V.PHẦN V : Cài đặt hệ thống

Từ 6 lược đồ ở dạng chuẩn 3 NF , ta xây dựng thành 6 bảng dữ liệu :  
\*6 BẢNG DỮ LIỆU

1 . Bảng Nhan\_vien : Thông tin về nhân viên công ty



2. Bảng Lam\_viec : Thông tin về ca làm việc của nhân viên theo ngày

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động

3. Bảng Xe : Thông tin về xe bus

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động

4. Bảng Diem\_dung : Thông tin về các điểm dừng xe bus

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động

5.Bảng Tuyen : Thông tin về các tuyến xe bus

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động

6.Bảng Lo\_trinh: Thông tin lộ trình các tuyến xe bus

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động

# \*\* CÁC CÂU TRUY VẤN DỮ LIỆU

[\*\* CÁC CÂU TRUY VẤN DỮ LIỆU 8](#_Toc169642574)

[1.Tìm , thống kê : 9](#_Toc169642575)

[+Liệt kê xem xe đi qua những điểm dừng nào 9](#_Toc169642576)

[+Liệt kê xem điểm này có những xe nào đi qua và các thông tin của xe, tần suất 9](#_Toc169642577)

[+Trên tuyến này có những xe nào chạy thời gian chạy, nhân viên của xe 9](#_Toc169642578)

[+Liệt kê tuyến này có những điểm dừng nào có bao nhiêu xe, tần suất 9](#_Toc169642579)

[+Tìm những nhân viên có tổng số ca làm việc lớn hơn 10 10](#_Toc169642580)

[2.Tính: 10](#_Toc169642581)

[+Tính tổng số ca, vé tháng và ngày bán được của  nhân viên 10](#_Toc169642582)

[+Tính số vé mà nhân viên bán trong ngày(2 loại) 10](#_Toc169642583)

[+Tính xem xe đã sử dụng được bao lâu 10](#_Toc169642584)

[+Tính số tuổi hiện tại của tất cả nhân viên 10](#_Toc169642585)

[+ Tính doanh thu của mỗi nhân viên trong 1 ngày 11](#_Toc169642586)

[3.View: 11](#_Toc169642587)

[Tạo View để người dùng xem thông tin xe, tuyến, thời gian chạy và các điểm dừng 11](#_Toc169642588)

[Tạo View để người dùng xem thông tin của nhân viên ID, Họ tên, Chức vụ, Giới tính, Ngày sinh 11](#_Toc169642589)

[4.Function: 12](#_Toc169642590)

[Hàm tính tổng số vé mỗi loại mà người đó bán 12](#_Toc169642591)

[Hàm tính tổng số ca họ làm việc trong ngày 12](#_Toc169642592)

[Hàm Tìm kiếm thông tin của xe, tuyến khi họ nhập vào nơi bắt đầu và nơi muốn đến 12](#_Toc169642593)

[Hàm nhập vào id nhân viên và ngày làm việc 13](#_Toc169642594)

[Hàm tính tổng số ca, tổng số vé mối loại của 1 nhân viên trong 1 tháng 14](#_Toc169642595)

[Hàm xem số ca mà một nhân viên làm việc trên 1 tuyến xe 14](#_Toc169642596)

[Hàm tìm tuyến xe dựa theo điểm dừng đầu tiên 15](#_Toc169642597)

[Hàm tìm tuyến xe, nhân viên, biển số, tên xe và hãng xe đi qua địa điểm 16](#_Toc169642598)

[5.Trigged: 17](#_Toc169642599)

[Viết Trigger khi Insert, Update, nhân viên vào 1 chức vụ xem đã đủ số người hay chưa 17](#_Toc169642600)

[Viết trigger check dữ liệu trước khi insert vào Lam\_viec 18](#_Toc169642601)

[Thêm cột tong\_tien vào bảng Lam\_viec 18](#_Toc169642602)

[Tạo thêm cột So\_diem\_dung trong bang Tuyen 19](#_Toc169642603)

[6.Index: 22](#_Toc169642604)

[Viết Index cho các câu lệnh tìm kiếm ở trên 22](#_Toc169642605)

## 1.Tìm , thống kê :

### +Liệt kê xem xe đi qua những điểm dừng nào

Câu lệnh SQL :

 select x."ten\_xe", d."Ten\_diem\_dung"

from quanly."Xe" as x

join quanly."Lam\_viec" as l on x."Bien\_so"=l."Bien\_so"

join quanly."Lo\_trinh" as lt on l."Ma\_tuyen"=lt."Ma\_tuyen"

join quanly."Diem\_dung" as d on lt."Ma\_diem\_dung"=d."Ma\_diem\_dung"

order by x."ten\_xe"

### +Liệt kê xem điểm này có những xe nào đi qua và các thông tin của xe, tần suất

Câu lệnh SQL :

select d."Ten\_diem\_dung", x."ten\_xe", t."Ten\_tuyen",t."Tan\_suat"

from quanly."Diem\_dung" as d

join quanly."Lo\_trinh" as l on d."Ma\_diem\_dung"=l."Ma\_diem\_dung"

join quanly."Tuyen" as t on l."Ma\_tuyen"=t."Ma\_tuyen"

join quanly."Lam\_viec" as lv on t."Ma\_tuyen"=lv."Ma\_tuyen"

join quanly."Xe" as x on lv."Bien\_so"=x."Bien\_so"

order by d."Ten\_diem\_dung"

### +Trên tuyến này có những xe nào chạy thời gian chạy, nhân viên của xe

Câu lệnh SQL :

select t."Ten\_tuyen", t."Gio\_bat\_dau", t."Gio\_ket\_thuc", x."ten\_xe", n."Ho\_va\_ten", n."Chuc\_vu"

from quanly."Tuyen" as t

join quanly."Lam\_viec" as l on t."Ma\_tuyen"=l."Ma\_tuyen"

join quanly."Xe" as x on l."Bien\_so"=x."Bien\_so"

join quanly."Nhan\_vien" as n on l."ID\_nhan\_vien"=n."ID\_nhan\_vien"

order by t."Ten\_tuyen"

### +Liệt kê tuyến này có những điểm dừng nào có bao nhiêu xe, tần suất

Câu lệnh SQL :

select t."Ten\_tuyen", d."Ten\_diem\_dung", t."So\_luong\_xe",t."Tan\_suat"

from quanly."Diem\_dung" as d

join quanly."Lo\_trinh" as l on d."Ma\_diem\_dung"=l."Ma\_diem\_dung"

join quanly."Tuyen" as t on l."Ma\_tuyen"=t."Ma\_tuyen"

order by t."Ten\_tuyen"

### +Tìm những nhân viên có tổng số ca làm việc lớn hơn 10

Câu lệnh SQL :

select n."Ho\_va\_ten", sum(l."So\_ca") as tong\_ca, l."Ngay\_lam\_viec"

from quanly."Nhan\_vien" as n

join quanly."Lam\_viec" as l on n."ID\_nhan\_vien"=l."ID\_nhan\_vien"

group by n."Ho\_va\_ten", l."Ngay\_lam\_viec"

having sum(l."So\_ca") > 10

## 2.Tính:

### +Tính tổng số ca, vé tháng và ngày bán được của  nhân viên

Câu lệnh SQL :

SELECT DATE(LamViec."Ngay\_lam\_viec"), NhanVien."Ho\_va\_ten",

SUM(LamViec."So\_ca") Tong\_so\_ca,

SUM(LamViec."So\_ve\_ngay") Tong\_so\_ve\_ngay,

SUM(LamViec."So\_ve\_thang") Tong\_so\_ve\_thang

FROM quanly."Nhan\_vien" NhanVien

JOIN quanly."Lam\_viec" LamViec on NhanVien."ID\_nhan\_vien" = LamViec."ID\_nhan\_vien"

GROUP BY NhanVien."Ho\_va\_ten", DATE(LamViec."Ngay\_lam\_viec")

### +Tính số vé mà nhân viên bán trong ngày(2 loại)

Câu lệnh SQL :

SELECT LamViec."Ngay\_lam\_viec", NhanVien."Ho\_va\_ten", NhanVien."Chuc\_vu",

SUM(LamViec."So\_ve\_ngay") Tong\_so\_ve\_ngay,

SUM(LamViec."So\_ve\_thang") Tong\_so\_ve\_thang

FROM quanly."Lam\_viec" LamViec, quanly."Nhan\_vien" NhanVien

WHERE LamViec."ID\_nhan\_vien" = NhanVien."ID\_nhan\_vien"

GROUP BY LamViec."ID\_nhan\_vien", NhanVien."Ho\_va\_ten", NhanVien."Chuc\_vu", LamViec."Ngay\_lam\_viec"

### +Tính xem xe đã sử dụng được bao lâu

Câu lệnh SQL :

SELECT Xe."Bien\_so", Xe."Hang\_xe", Xe."ten\_xe",

CURRENT\_DATE - Xe."Nam\_sx" AS So\_ngay\_su\_dung

FROM quanly."Xe" Xe;

### +Tính số tuổi hiện tại của tất cả nhân viên

Câu lệnh SQL :

SELECT NhanVien."ID\_nhan\_vien", NhanVien."Ho\_va\_ten", NhanVien."Chuc\_vu",

DATE\_PART('year', CURRENT\_DATE) - DATE\_PART('year', NhanVien."Ngay\_sinh") Tuoi

from quanly."Nhan\_vien" NhanVien

### + Tính doanh thu của mỗi nhân viên trong 1 ngày

(số vé ngày \* 3000, số vé tháng \* 30000)

Câu lệnh SQL :

select l."ID\_nhan\_vien", n."Ho\_va\_ten", sum(l."So\_ve\_ngay")\*3000 + sum(l."So\_ve\_thang")\*30000 as Doanh\_thu, l."Ngay\_lam\_viec"

from quanly."Lam\_viec" as l

join quanly."Nhan\_vien" as n on l."ID\_nhan\_vien"=n."ID\_nhan\_vien"

group by l."ID\_nhan\_vien", n."Ho\_va\_ten",l."Ngay\_lam\_viec"

order by sum(l."So\_ve\_ngay")\*3000 + sum(l."So\_ve\_thang")\*30000

## 3.View:

### Tạo View để người dùng xem thông tin xe, tuyến, thời gian chạy và các điểm dừng

Câu lệnh SQL :

create or replace view quanly."TTXe" AS

select x."ten\_xe",t."Gio\_bat\_dau", t."Gio\_ket\_thuc", t."Tan\_suat", d."Ten\_diem\_dung"

from quanly."Xe" as x join quanly."Lam\_viec" as l on x."Bien\_so"=l."Bien\_so"

join quanly."Tuyen" as t on l."Ma\_tuyen"=t."Ma\_tuyen"

join quanly."Lo\_trinh" as lt on t."Ma\_tuyen"=lt."Ma\_tuyen"

join quanly."Diem\_dung" as d on lt."Ma\_diem\_dung"=d."Ma\_diem\_dung"

### Tạo View để người dùng xem thông tin của nhân viên ID, Họ tên, Chức vụ, Giới tính, Ngày sinh

Câu lệnh SQL :

create view quanly."TTNhan\_vien" As

select "ID\_nhan\_vien","Ho\_va\_ten","Gioi\_tinh","Chuc\_vu", "Ngay\_sinh"

from quanly."Nhan\_vien"

## 4.Function:

### Hàm tính tổng số vé mỗi loại mà người đó bán

Câu lệnh SQL :

CREATE OR REPLACE FUNCTION SoVeNhanVienBanDuoc(IN \_inputID Int4)

RETURNS TABLE(id\_nhan\_vien Int4, ngay\_lam\_viec Timestamp, so\_ve\_ngay BigInt, so\_ve\_thang BigInt)

AS $$

BEGIN

RETURN QUERY

SELECT LamViec."ID\_nhan\_vien", LamViec."Ngay\_lam\_viec",

SUM(LamViec."So\_ve\_ngay"), SUM(LamViec."So\_ve\_thang")

FROM quanly."Lam\_viec" LamViec

WHERE \_inputID = LamViec."ID\_nhan\_vien"

GROUP BY LamViec."ID\_nhan\_vien", LamViec."Ngay\_lam\_viec";

END; $$

LANGUAGE plpgsql;

SELECT \* FROM SoVeNhanVienBanDuoc(20009637)

### Hàm tính tổng số ca họ làm việc trong ngày

Câu lệnh SQL :

CREATE OR REPLACE FUNCTION SoCaLamViec(IN \_inputID Int4)

RETURNS TABLE (ngay\_lam\_viec TimeStamp, id\_nhan\_vien Int4, tong\_so\_ca BigInt)

AS $$

BEGIN

RETURN QUERY

SELECT LamViec."Ngay\_lam\_viec", LamViec."ID\_nhan\_vien", SUM(LamViec."So\_ca")

FROM quanly."Lam\_viec" LamViec

WHERE \_inputID = LamViec."ID\_nhan\_vien"

GROUP BY LamViec."ID\_nhan\_vien", LamViec."Ngay\_lam\_viec";

END; $$

LANGUAGE plpgsql;

SELECT \* FROM SoCaLamViec(20003857)

### Hàm Tìm kiếm thông tin của xe, tuyến khi họ nhập vào nơi bắt đầu và nơi muốn đến

Câu lệnh SQL :

create or replace function TTim\_xe(A varchar, B varchar)

returns table (Ten\_xe varchar, Ten\_tuyen varchar, Tan\_suat varchar) AS $$

begin

return query select x."ten\_xe", t."Ten\_tuyen", t."Tan\_suat"

from quanly."Diem\_dung" as d

join quanly."Lo\_trinh" as l on d."Ma\_diem\_dung"=l."Ma\_diem\_dung"

join quanly."Tuyen" as t on l."Ma\_tuyen"=t."Ma\_tuyen"

join quanly."Lam\_viec" as lv on t."Ma\_tuyen"=lv."Ma\_tuyen"

join quanly."Xe" as x on lv."Bien\_so"=x."Bien\_so"

where A=d."Ten\_diem\_dung"

intersect

select x."ten\_xe", t."Ten\_tuyen", t."Tan\_suat"

from quanly."Diem\_dung" as d

join quanly."Lo\_trinh" as l on d."Ma\_diem\_dung"=l."Ma\_diem\_dung"

join quanly."Tuyen" as t on l."Ma\_tuyen"=t."Ma\_tuyen"

join quanly."Lam\_viec" as lv on t."Ma\_tuyen"=lv."Ma\_tuyen"

join quanly."Xe" as x on lv."Bien\_so"=x."Bien\_so"

where B=d."Ten\_diem\_dung";

end;

$$ language plpgsql;

select \* from TTim\_xe('Từ Sơn - Quốc Lộ 1', 'Cổng Parabol Đh Bách Khoa')

### Hàm nhập vào id nhân viên và ngày làm việc

( Trả về thông tin liên quan đến nhân viên đó trong ngày hôm đó)

create or replace function TT\_nhan\_vien(A integer, B timestamp)

returns table(ID\_NV int,Ten\_NV varchar, Chuc\_vu varchar, So\_ca bigint, So\_ve\_ngay bigint, So\_ve\_thang bigint, Ten\_xe varchar, Ten\_tuyen varchar) as $$

begin

return query select n."ID\_nhan\_vien", n."Ho\_va\_ten", n."Chuc\_vu", sum(l."So\_ca"), sum(l."So\_ve\_ngay"), sum(l."So\_ve\_thang"),x."ten\_xe", t."Ten\_tuyen"

from quanly."Nhan\_vien" as n

join quanly."Lam\_viec" as l on n."ID\_nhan\_vien"=l."ID\_nhan\_vien"

join quanly."Xe" as x on l."Bien\_so"=x."Bien\_so"

join quanly."Tuyen" as t on l."Ma\_tuyen"=t."Ma\_tuyen"

where n."ID\_nhan\_vien"=A and l."Ngay\_lam\_viec"=B

group by n."ID\_nhan\_vien", n."Ho\_va\_ten", n."Chuc\_vu",x."ten\_xe", t."Ten\_tuyen";

end;

$$ language plpgsql;

select \* from TT\_nhan\_vien(20010664, '2024-05-14')

### Hàm tính tổng số ca, tổng số vé mối loại của 1 nhân viên trong 1 tháng

CREATE OR REPLACE FUNCTION TTCong\_Thang(A INT, B INT, C INT)

RETURNS TABLE(Ma\_nv INT, Tong\_ca BIGINT, Tong\_ve\_ngay BIGINT, Tong\_ve\_thang BIGINT, thang INT, nam INT) AS $$

BEGIN

RETURN QUERY SELECT n."ID\_nhan\_vien", SUM(l."So\_ca"), SUM(l."So\_ve\_ngay"), SUM(l."So\_ve\_thang"), EXTRACT(MONTH FROM l."Ngay\_lam\_viec")::INT, EXTRACT(YEAR FROM l."Ngay\_lam\_viec")::INT

FROM quanly."Nhan\_vien" AS n

JOIN quanly."Lam\_viec" AS l ON n."ID\_nhan\_vien" = l."ID\_nhan\_vien"

WHERE n."ID\_nhan\_vien" = A AND

B = EXTRACT(MONTH FROM l."Ngay\_lam\_viec") AND

C = EXTRACT(YEAR FROM l."Ngay\_lam\_viec")

GROUP BY n."ID\_nhan\_vien", EXTRACT(MONTH FROM l."Ngay\_lam\_viec"), EXTRACT(YEAR FROM l."Ngay\_lam\_viec");

END;

$$ LANGUAGE plpgsql;

select \* from TTCOng\_Thang(20010500, 05, 2024)

### Hàm xem số ca mà một nhân viên làm việc trên 1 tuyến xe

CREATE OR REPLACE FUNCTION SoCaNhanVienLamDuocTrenMotTuyenXe()

RETURNS TABLE(id\_nhan\_vien Int4, ten\_nhan\_vien VarChar, so\_ca Int4, ngay\_lam\_viec TimeStamp, ten\_xe VarChar)

AS $$

BEGIN

RETURN QUERY

SELECT NhanVien."ID\_nhan\_vien", NhanVien."Ho\_va\_ten", LamViec."So\_ca", LamViec."Ngay\_lam\_viec", Xe."ten\_xe"

FROM quanly."Nhan\_vien" NhanVien

LEFT JOIN quanly."Lam\_viec" LamViec

ON NhanVien."ID\_nhan\_vien" = LamViec."ID\_nhan\_vien"

LEFT JOIN quanly."Xe" Xe

ON LamViec."Bien\_so" = Xe."Bien\_so"

WHERE Xe."ten\_xe" IS NOT NULL;

END; $$

LANGUAGE plpgsql;

SELECT \* FROM SoCaNhanVienLamDuocTrenMotTuyenXe();

### Hàm tìm tuyến xe dựa theo điểm dừng đầu tiên

CREATE OR REPLACE FUNCTION TimTuyenXeTheoDiemDungDauTien(IN diem\_dung\_dau\_tiens VarChar)

RETURNS TABLE(ma\_tuyen VarChar, ten\_tuyen VarChar)

AS $$

BEGIN

RETURN QUERY

SELECT Bang."Ma\_tuyen", Bang."Ten\_tuyen" FROM

(SELECT Tuyen."Ma\_tuyen", Tuyen."Ten\_tuyen", DiemDung."Ten\_diem\_dung" AS "Diem\_dung\_dau\_tien"

FROM quanly."Tuyen" AS Tuyen

JOIN

(SELECT "Ma\_tuyen", MIN("Ma\_diem\_dung") AS "Ma\_diem\_dung\_dau\_tien"

FROM quanly."Lo\_trinh"

GROUP BY "Ma\_tuyen") LoTrinh

ON Tuyen."Ma\_tuyen" = LoTrinh."Ma\_tuyen"

JOIN quanly."Diem\_dung" DiemDung ON LoTrinh."Ma\_diem\_dung\_dau\_tien" = DiemDung."Ma\_diem\_dung") AS Bang

WHERE Bang."Diem\_dung\_dau\_tien" = diem\_dung\_dau\_tiens;

END; $$

LANGUAGE plpgsql;

SELECT \* FROM TimTuyenXeTheoDiemDungDauTien('Từ Sơn - Quốc Lộ 1');

### Hàm tìm tuyến xe, nhân viên, biển số, tên xe và hãng xe đi qua địa điểm

CREATE OR REPLACE FUNCTION TimNhanVienVaXeDiQuaDiaDiem(IN dia\_diem VarChar)

RETURNS TABLE(ten\_tuyen VarChar, ten\_nhan\_vien VarChar, bien\_so VarChar, ten\_xe VarChar, hang\_xe VarChar)

AS $$

BEGIN

RETURN QUERY

SELECT Tuyen."Ten\_tuyen" , Nhan\_vien."Ho\_va\_ten" , Xe."Bien\_so" , Xe."ten\_xe" , Xe."Hang\_xe"

FROM quanly."Diem\_dung" Diem\_dung INNER JOIN quanly."Lo\_trinh" Lo\_trinh

ON Diem\_dung."Ma\_diem\_dung" = Lo\_trinh."Ma\_diem\_dung"

INNER JOIN quanly."Tuyen" Tuyen ON Lo\_trinh."Ma\_tuyen" = Tuyen."Ma\_tuyen"

INNER JOIN quanly."Lam\_viec" Lam\_viec ON Tuyen."Ma\_tuyen" = Lam\_viec."Ma\_tuyen"

INNER JOIN quanly."Nhan\_vien" Nhan\_vien ON Lam\_viec."ID\_nhan\_vien" = Nhan\_vien."ID\_nhan\_vien"

INNER JOIN quanly."Xe" Xe ON Lam\_viec."Bien\_so" = Xe."Bien\_so"

WHERE Diem\_dung."Dia\_diem" = dia\_diem;

END; $$

LANGUAGE plpgsql;

SELECT \* FROM TimNhanVienVaXeDiQuaDiaDiem('185 Minh Khai ');

## 5.Trigged:

### Viết Trigger khi Insert, Update, nhân viên vào 1 chức vụ xem đã đủ số người hay chưa

Câu lệnh SQL :

 create or replace trigger check\_NV

before insert or update on quanly."Nhan\_vien"

for each row

execute procedure check\_number\_of\_NV();

create or replace function check\_number\_of\_NV()

returns trigger as $$

declare num\_NV bigint;

begin

select count(n."ID\_nhan\_vien") INTO num\_NV

from quanly."Nhan\_vien" as n

where n."Chuc\_vu"=new."Chuc\_vu";

if num\_NV >= 2501 then

raise exception 'Da du nhan vien cho vi tri nay';

end if;

return new;

end;

$$ language plpgsql;

insert into quanly."Nhan\_vien"("Ho\_va\_ten", "ID\_nhan\_vien","Chuc\_vu","Gioi\_tinh") values('ABCDE',29999999,'NV bán vé','M')

### Viết trigger check dữ liệu trước khi insert vào Lam\_viec

create or replace function check\_lamviec()

returns trigger as $$

begin

if new."So\_ca" <0 or new."So\_ve\_ngay" <0 or new."So\_ve\_thang" <0 then

raise exception 'Loi du lieu';

end if;

return new;

end;

$$ language plpgsql;

create or replace trigger check\_lv

before insert on quanly."Lam\_viec"

for each row

execute procedure check\_lamviec();

insert into quanly."Lam\_viec" values(20004957, 'HN2001', NULL, -2, 3,5, NULL)

### Thêm cột tong\_tien vào bảng Lam\_viec

Tong\_tien= So\_ve\_ngay\*7000 + So\_ve\_thang\* 100000

Update cột và viết trigger tự tính cột trên khi insert hoặc uppdate vào bảng

create or replace function TTong\_tien()

returns trigger as $$

begin

NEW."tong\_tien" := NEW."So\_ve\_ngay" \* 7000 + NEW."So\_ve\_thang" \* 100000;

return new;

end;

$$ language plpgsql;

create or replace trigger TTong\_tien1

before insert or update on quanly."Lam\_viec"

for each row

execute function TTong\_tien();

insert into quanly."Lam\_viec"("ID\_nhan\_vien","Ma\_tuyen","Bien\_so","So\_ca","So\_ve\_ngay","So\_ve\_thang") values(20004957, 'HN2001', NULL, 5, 3, 3)

### Tạo thêm cột So\_diem\_dung trong bang Tuyen

( Tạo Trigger cột trên và tự động cập nhật lại Số điểm dừng của tuyên khi insert, update, delate)

alter table quanly."Tuyen"

add column "So\_diem\_dung" bigint;

UPDATE quanly."Tuyen" t

SET "So\_diem\_dung" = l.dem\_diem\_dung

FROM (

SELECT "Ma\_tuyen", COUNT("Ma\_diem\_dung") AS dem\_diem\_dung

FROM quanly."Lo\_trinh"

GROUP BY "Ma\_tuyen"

) l

WHERE t."Ma\_tuyen" = l."Ma\_tuyen";

CREATE OR REPLACE FUNCTION Auto\_TTuyen()

RETURNS TRIGGER AS $$

BEGIN

IF TG\_OP = 'INSERT' THEN

UPDATE quanly."Tuyen" t

SET "So\_diem\_dung" = t."So\_diem\_dung" + 1

WHERE t."Ma\_tuyen" = NEW."Ma\_tuyen";

ELSIF TG\_OP = 'DELETE' THEN

UPDATE quanly."Tuyen" t

SET "So\_diem\_dung" = t."So\_diem\_dung" - 1

WHERE t."Ma\_tuyen" = OLD."Ma\_tuyen";

ELSIF TG\_OP = 'UPDATE' THEN

IF OLD."Ma\_tuyen" <> NEW."Ma\_tuyen" THEN

UPDATE quanly."Tuyen" t

SET "So\_diem\_dung" = t."So\_diem\_dung" - 1

WHERE t."Ma\_tuyen" = OLD."Ma\_tuyen";

UPDATE quanly."Tuyen" t

SET "So\_diem\_dung" = t."So\_diem\_dung" + 1

WHERE t."Ma\_tuyen" = NEW."Ma\_tuyen";

END IF;

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER Auto\_TTuyen1

AFTER INSERT OR DELETE OR UPDATE ON quanly."Lo\_trinh"

FOR EACH ROW

EXECUTE FUNCTION Auto\_TTuyen();

CREATE OR REPLACE FUNCTION Auto\_TTuyen()

RETURNS TRIGGER AS $$

BEGIN

IF TG\_OP = 'INSERT' THEN

UPDATE quanly."Tuyen" t

SET "So\_diem\_dung" = t."So\_diem\_dung" + 1

WHERE t."Ma\_tuyen" = NEW."Ma\_tuyen";

ELSIF TG\_OP = 'DELETE' THEN

UPDATE quanly."Tuyen" t

SET "So\_diem\_dung" = t."So\_diem\_dung" - 1

WHERE t."Ma\_tuyen" = OLD."Ma\_tuyen";

ELSIF TG\_OP = 'UPDATE' THEN

IF OLD."Ma\_tuyen" <> NEW."Ma\_tuyen" THEN

UPDATE quanly."Tuyen" t

SET "So\_diem\_dung" = t."So\_diem\_dung" - 1

WHERE t."Ma\_tuyen" = OLD."Ma\_tuyen";

UPDATE quanly."Tuyen" t

SET "So\_diem\_dung" = t."So\_diem\_dung" + 1

WHERE t."Ma\_tuyen" = NEW."Ma\_tuyen";

END IF;

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER Auto\_TTuyen1

AFTER INSERT OR DELETE OR UPDATE ON quanly."Lo\_trinh"

FOR EACH ROW

EXECUTE FUNCTION Auto\_TTuyen();

## 6.Index:

### Liệt kê xe nào đi qua 'Điểm dừng số 260’

CREATE INDEX idx\_Ten\_diem\_dung ON quanly."Diem\_dung" USING hash ("Ten\_diem\_dung");

EXPLAIN ANALYZE select x."ten\_xe", d."Ten\_diem\_dung"

from quanly."Xe" as x

join quanly."Lam\_viec" as l on x."Bien\_so"=l."Bien\_so"

join quanly."Lo\_trinh" as lt on l."Ma\_tuyen"=lt."Ma\_tuyen"

join quanly."Diem\_dung" as d on lt."Ma\_diem\_dung"=d."Ma\_diem\_dung"

where d."Ten\_diem\_dung"='Điểm dừng số 260'

order by x."ten\_xe"





### Liệt kê xe 3286 đi qua những điểm nào

CREATE INDEX idx\_Ten\_xe ON quanly."Xe" USING hash ("ten\_xe");

EXPLAIN ANALYZE select d."Ten\_diem\_dung", x."ten\_xe", t."Ten\_tuyen",t."Tan\_suat"

from quanly."Diem\_dung" as d

join quanly."Lo\_trinh" as l on d."Ma\_diem\_dung"=l."Ma\_diem\_dung"

join quanly."Tuyen" as t on l."Ma\_tuyen"=t."Ma\_tuyen"

join quanly."Lam\_viec" as lv on t."Ma\_tuyen"=lv."Ma\_tuyen"

join quanly."Xe" as x on lv."Bien\_so"=x."Bien\_so"

where x."ten\_xe"='3286'

order by d."Ten\_diem\_dung"





### Liệt ke tuyến Giáp Bát-Bắc Từ Liêm có những xe nào chạy thời gian chạy

CREATE INDEX idx\_Ten\_tuyen ON quanly."Tuyen" USING hash ("Ten\_tuyen");

EXPLAIN ANALYZE select t."Ten\_tuyen", t."Gio\_bat\_dau", t."Gio\_ket\_thuc", x."ten\_xe", n."Ho\_va\_ten", n."Chuc\_vu"

from quanly."Tuyen" as t

join quanly."Lam\_viec" as l on t."Ma\_tuyen"=l."Ma\_tuyen"

join quanly."Xe" as x on l."Bien\_so"=x."Bien\_so"

join quanly."Nhan\_vien" as n on l."ID\_nhan\_vien"=n."ID\_nhan\_vien"

where t."Ten\_tuyen"='Giáp Bát-Bắc Từ Liêm'

order by t."Ten\_tuyen"





### Liệt ke tuyến Giáp Bát-Bắc Từ Liêm có những điểm dừng nào

CREATE INDEX idx\_Ten\_tuyen ON quanly."Tuyen" USING hash ("Ten\_tuyen");

EXPLAIN ANALYZE select t."Ten\_tuyen", d."Ten\_diem\_dung", t."So\_luong\_xe",t."Tan\_suat"

from quanly."Diem\_dung" as d

join quanly."Lo\_trinh" as l on d."Ma\_diem\_dung"=l."Ma\_diem\_dung"

join quanly."Tuyen" as t on l."Ma\_tuyen"=t."Ma\_tuyen"

where t."Ten\_tuyen" ='Giáp Bát-Bắc Từ Liêm'

order by t."Ten\_tuyen"





### Liệt kê các nhân viên có số ca làm >3

CREATE INDEX idx\_So\_ca ON quanly."Lam\_viec" USING btree ("So\_ca");

EXPLAIN ANALYZE select n."Ho\_va\_ten", l."So\_ca", l."Ngay\_lam\_viec"

from quanly."Nhan\_vien" as n

join quanly."Lam\_viec" as l on n."ID\_nhan\_vien"=l."ID\_nhan\_vien"

where l."So\_ca" > 3



