**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**Trường CNTT & TT**

**Báo cáo cuối kỳ môn Thực hành Cơ**

**sở dữ liệu**

**Giáo viên phụ trách: Vũ Tuyết Trinh**

**Nhóm 7 - Lớp: 147775**

**Hệ thống quản lý dữ liệu dân cư và dịch vụ khu chung cư**

**Nguyễn Khắc Hiếu 20225839**

**Đinh Đình Lộc 20225647**

**I.** **Mô tả bài toán**

1. Title:

Hệ thống quản lý dữ liệu dân cư và dịch vụ khu chung cư.

2. Context:

Xây dựng cơ sở dữ liệu nhằm quản lý thông tin của toàn bộ dân cư và khách hàng sử dụng tiện ích trong tòa nhà.

3. Description:

- Hệ thống cho phép ban quản trị (admin) quản lý thông tin cá nhân của cư dân, thông tin liên lạc, thông tin về căn hộ (diện tích, số phòng) và các thông tin khác liên quan.

- Cho phép cư dân cập nhật thông tin của mình bao gồm thông tin cá nhân và thông tin sử dụng các dịch vụ tiện ích.

- Quản lý hệ thống điện, nước sử dụng và các loại phí dịch vụ hàng tháng của từng căn hộ.

- Quản lý thông tin khách hàng sử dụng tiện ích.

4. Requirements:

- Admin có thể thêm, xóa, chỉnh sửa, tìm kiếm thông tin của từng hộ dân.

- Cư dân có thể chỉnh sửa thông tin căn hộ của mình (bao gồm thông tin về người ở, thông tin liên lạc, phương tiện đăng ký) và xem thông tin liên lạc cơ bản của admin.

- Cư dân đăng ký, hủy bỏ việc sử dụng các tiện ích như hầm bãi, bể bơi, phòng tập (thông tin cá nhân, có phải người ở chung cư hay không, thời hạn sử dụng dịch vụ đăng ký, lưu lại hóa đơn đã thanh toán).

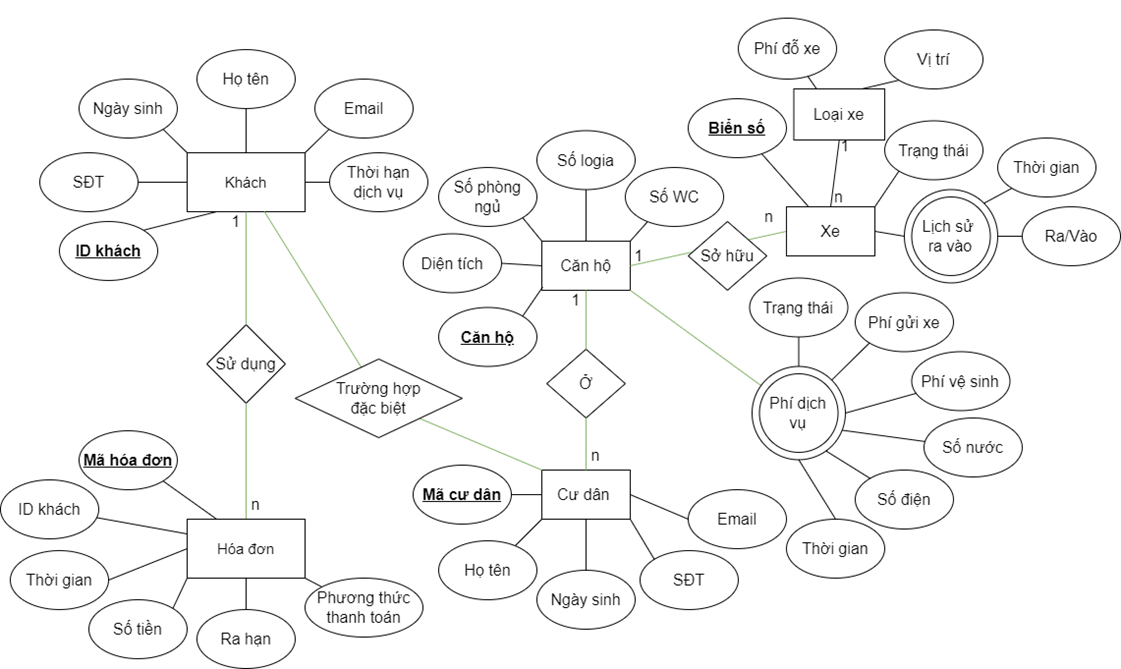
- Hệ thống này quản lý việc tạo và gửi hóa đơn dịch vụ hàng tháng của từng hộ dân (dựa trên diện tích căn hộ, số người ở, số lượng xe và loại xe) cũng như việc thu tiền và cập nhật trạng thái thanh toán.

- Cập nhật thông tin sử dụng điện, nước của từng căn hộ hàng tháng để đối chiếu với hóa đơn điện, nước.

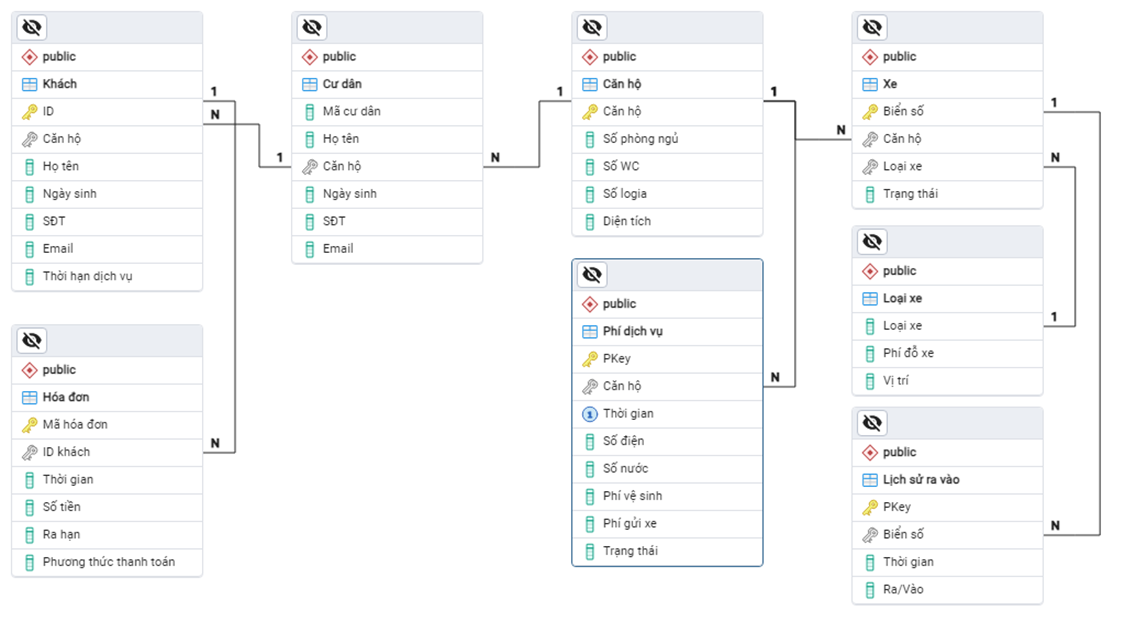
- Kiểm soát xe ra vào hầm đỗ.

**II.** **Thiết kế CSDL:**

1. Sơ đồ thực thể - liên kết:



2. Sơ đồ quan hệ:



**III.** **Các chức năng thực hiện:**

1. Truy vấn SQL để tạo ra các bảng và ràng buộc:

BEGIN;

CREATE TABLE IF NOT EXISTS public."Cư dân"

(

"Mã cư dân" bigserial NOT NULL,

"Họ tên" character varying NOT NULL,

"Căn hộ" character varying NOT NULL,

"Ngày sinh" date NOT NULL,

"SĐT" character varying NOT NULL,

"Email" character varying NOT NULL

);

CREATE TABLE IF NOT EXISTS public."Căn hộ"

(

"Căn hộ" character varying NOT NULL,

"Số phòng ngủ" integer NOT NULL,

"Số WC" integer NOT NULL,

"Số logia" integer NOT NULL,

"Diện tích" double precision NOT NULL,

PRIMARY KEY ("Căn hộ")

);

CREATE TABLE IF NOT EXISTS public."Xe"

(

"Biển số" character varying NOT NULL,

"Căn hộ" character varying NOT NULL,

"Loại xe" character varying NOT NULL,

"Trạng thái" character varying NOT NULL,

PRIMARY KEY ("Biển số")

);

CREATE TABLE IF NOT EXISTS public."Phí dịch vụ"

(

"PKey" bigserial NOT NULL,

"Căn hộ" character varying NOT NULL,

"Thời gian" date NOT NULL,

"Số điện" integer NOT NULL,

"Số nước" integer NOT NULL,

"Phí vệ sinh" integer NOT NULL,

"Phí gửi xe" integer NOT NULL,

"Trạng thái" character varying NOT NULL,

PRIMARY KEY ("PKey"),

UNIQUE ("Căn hộ", "Thời gian")

);

CREATE TABLE IF NOT EXISTS public."Khách"

(

"ID" bigserial NOT NULL,

"Căn hộ" character varying,

"Họ tên" character varying NOT NULL,

"Ngày sinh" date NOT NULL,

"SĐT" character varying NOT NULL,

"Email" character varying NOT NULL,

"Thời hạn dịch vụ" date NOT NULL,

PRIMARY KEY ("ID")

);

CREATE TABLE IF NOT EXISTS public."Hóa đơn"

(

"Mã hóa đơn" bigserial NOT NULL,

"ID khách" integer NOT NULL,

"Thời gian" date NOT NULL,

"Số tiền" integer NOT NULL,

"Ra hạn" date NOT NULL,

"Phương thức thanh toán" character varying NOT NULL,

PRIMARY KEY ("Mã hóa đơn")

);

CREATE TABLE IF NOT EXISTS public."Lịch sử ra vào"

(

"PKey" bigserial,

"Biển số" character varying NOT NULL,

"Thời gian" time with time zone NOT NULL,

"Ra/Vào" character varying NOT NULL,

PRIMARY KEY ("PKey")

);

CREATE TABLE IF NOT EXISTS public."Loại xe"

(

"Loại xe" character varying NOT NULL,

"Phí đỗ xe" integer NOT NULL,

"Vị trí" character varying NOT NULL

);

ALTER TABLE IF EXISTS public."Cư dân"

ADD FOREIGN KEY ("Căn hộ")

REFERENCES public."Căn hộ" ("Căn hộ") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID;

ALTER TABLE IF EXISTS public."Xe"

ADD CONSTRAINT "Căn hộ" FOREIGN KEY ("Căn hộ")

REFERENCES public."Căn hộ" ("Căn hộ") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID;

ALTER TABLE IF EXISTS public."Xe"

ADD CONSTRAINT "Loại xe" FOREIGN KEY ("Loại xe")

REFERENCES public."Loại xe" ("Loại xe") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID;

ALTER TABLE IF EXISTS public."Phí dịch vụ"

ADD CONSTRAINT "Căn hộ" FOREIGN KEY ("Căn hộ")

REFERENCES public."Căn hộ" ("Căn hộ") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID;

ALTER TABLE IF EXISTS public."Khách"

ADD FOREIGN KEY ("Căn hộ")

REFERENCES public."Cư dân" ("Căn hộ") MATCH SIMPLE

ON UPDATE SET NULL

ON DELETE SET NULL

NOT VALID;

ALTER TABLE IF EXISTS public."Hóa đơn"

ADD CONSTRAINT "ID Khách" FOREIGN KEY ("ID khách")

REFERENCES public."Khách" ("ID") MATCH SIMPLE

ON UPDATE SET NULL

ON DELETE SET NULL

NOT VALID;

ALTER TABLE IF EXISTS public."Lịch sử ra vào"

ADD CONSTRAINT "Biển số" FOREIGN KEY ("Biển số")

REFERENCES public."Xe" ("Biển số") MATCH SIMPLE

ON UPDATE SET NULL

ON DELETE SET NULL

NOT VALID;

END;

2. Các function:

**Thêm cư dân:**

CREATE OR REPLACE FUNCTION add\_cu\_dan("Họ tên" text,

"Căn hộ" text,

"Ngày sinh" date,

"SĐT" text,

"Email" text)

RETURNS void AS $$

BEGIN

INSERT INTO public."Cư dân"("Họ tên", "Căn hộ","Ngày sinh", "SĐT", "Email")

VALUES ("Họ tên", "Căn hộ","Ngày sinh", "SĐT", "Email");

END;

$$ LANGUAGE plpgsql;

**Thêm xe:**

CREATE OR REPLACE FUNCTION add\_xe("Biển số" text,

"Căn hộ" text,

"Loại xe" text )

RETURNS void AS $$

BEGIN

INSERT INTO public."Xe"("Biển số", "Căn hộ", "Loại xe","Trạng thái")

VALUES("Biển số", "Căn hộ", "Loại xe", 'In');

END;

$$ LANGUAGE plpgsql;

**Thêm khách:**

CREATE OR REPLACE FUNCTION add\_khach("Căn hộ" text,

"Họ tên" text,

"Ngày sinh" date,

"SĐT" text,

"Email" text,

"Thời hạn dịch vụ" date)

RETURNS void AS $$

BEGIN

INSERT INTO public."Khách"("Căn hộ","Họ tên","Ngày sinh","SĐT", "Email","Thời hạn dịch vụ")

VALUES ("Căn hộ","Họ tên","Ngày sinh","SĐT", "Email","Thời hạn dịch vụ");

exception when unique\_violation then raise notice 'Đã tồn tại khách hàng này!';

END;

$$ LANGUAGE plpgsql;

**Thêm hóa đơn:**

CREATE OR REPLACE FUNCTION add\_hoa\_don(ten TEXT,

sdt text,

so\_tien integer,

ra\_han date,

phuong\_thuc\_thanh\_toan text)

RETURNS VOID AS $$

declare

id int;

thoi\_gian date;

BEGIN

select "ID" from "Khách" into id

where ten="Họ tên" and sdt="SĐT";

if id is null

then

raise notice 'Không có khách hàng này!';

return;

end if;

select now() into thoi\_gian;

if ra\_han < thoi\_gian then raise notice 'Ra hạn không hợp lệ!'; return; end if;

update "Khách" set "Thời hạn dịch vụ"=ra\_han where "Khách"."ID"=id;

insert into "Hóa đơn"("ID khách", "Thời gian", "Số tiền", "Ra hạn", "Phương thức thanh toán")

values (id, thoi\_gian, so\_tien, ra\_han, phuong\_thuc\_thanh\_toan);

END;

$$ LANGUAGE plpgsql;

**Xóa cư dân:**

CREATE OR REPLACE FUNCTION delete\_cu\_dan(ho\_ten text,

can\_ho text)

RETURNS void AS $$

BEGIN

delete from "Cư dân"

where "Họ tên" = ho\_ten and "Căn hộ" = can\_ho;

END;

$$ LANGUAGE plpgsql;

**Xóa xe:**

CREATE OR REPLACE FUNCTION delete\_xe(bien\_so text)

RETURNS void AS $$

BEGIN

DELETE FROM "Xe" WHERE "Biển số" = bien\_so;

END;

$$ LANGUAGE plpgsql;

**Xóa khách dùng dịch vụ:**

CREATE OR REPLACE FUNCTION delete\_khach(ten text, sdt text)

RETURNS void AS $$

BEGIN

delete from "Khách" where "Họ tên" = ten and "SĐT" = sdt;

END;

$$ LANGUAGE plpgsql;

**Cập nhật thông tin cư dân:**

CREATE OR REPLACE FUNCTION update\_cu\_dan(old\_ho\_ten text,

old\_room text,

new\_ho\_ten text,

new\_can\_ho text,

ngay\_sinh date,

sdt text,

email text)

RETURNS void AS $$

declare ma\_cu\_dan int;

BEGIN

select "Mã cư dân" from "Cư dân" into ma\_cu\_dan

WHERE "Họ tên"=old\_ho\_ten and "Căn hộ"=room ;

if new\_ho\_ten is not null

then

update "Cư dân" set "Họ tên"=new\_ho\_ten where "Mã cư dân"=ma\_cu\_dan;

end if;

if new\_can\_ho is not null

then

update "Cư dân" set "Căn hộ"=new\_can\_ho where "Mã cư dân"=ma\_cu\_dan;

end if;

if ngay\_sinh is not null

then

update "Cư dân" set "Ngày sinh"=ngay\_sinh where "Mã cư dân"=ma\_cu\_dan;

end if;

if sdt is not null

then

update "Cư dân" set "SĐT"=sdt where "Mã cư dân"=ma\_cu\_dan;

end if;

if email is not null

then

update "Cư dân" set "Email"=email where "Mã cư dân"=ma\_cu\_dan;

end if;

END;

$$ LANGUAGE plpgsql;

**Cập nhật thông tin xe:**

CREATE OR REPLACE FUNCTION update\_xe(old\_bien\_xe text,

new\_bien\_xe text,

can\_ho text,

loai\_xe text)

RETURNS void AS $$

declare ma\_cu\_dan int;

BEGIN

if new\_bien\_xe is not null

then

update "Xe" set "Biển số"=new\_bien\_xe

where "Biển số"=old\_bien\_xe;

end if;

if can\_ho is not null

then

update "Xe" set "Căn hộ"=can\_ho

where "Biển số"=old\_bien\_xe;

end if;

if loai\_xe is not null

then

update "Xe" set "Loại xe"=loai\_xe

where "Biển số"=old\_bien\_xe;

end if;

END;

$$ LANGUAGE plpgsql;

**Cập nhật phí đỗ xe:**

CREATE OR REPLACE FUNCTION update\_phi\_doxe(loai\_xe text, tien integer)

RETURNS void AS $$

BEGIN

update "Loại xe" set "Phí đỗ xe"=tien where "Loại xe"=loai\_xe;

END;

$$ LANGUAGE plpgsql;

**Cập nhật thông tin phí dịch vụ:**

CREATE OR REPLACE FUNCTION update\_phi\_dv(can\_ho text,

thang integer,

so\_dien integer,

so\_nuoc integer,

phi\_vs integer,

phi\_xe integer)

RETURNS void AS $$

declare ma\_cu\_dan int;

declare id int;

BEGIN

select "PKey" into id from "Phí dịch vụ"

where "Căn hộ"=can\_ho

and thang = (select EXTRACT(month FROM "Thời gian"))

and (select EXTRACT(year FROM "Thời gian"))=(select EXTRACT(year FROM now()));

if id is null then raise notice 'Không có thông tin!'; return; end if;

if so\_dien is not null

then

update "Phí dịch vụ" set "Số điện"=so\_dien where "PKey"=id;

end if;

if so\_nuoc is not null

then

update "Phí dịch vụ" set "Số nước"=so\_nuoc where "PKey"=id;

end if;

if phi\_vs is not null

then

update "Phí dịch vụ" set "Phí vệ sinh"=phi\_vs where "PKey"=id;

end if;

if phi\_xe is not null

then

update "Phí dịch vụ" set "Phí gửi xe"=phi\_xe where "PKey"=id;

end if;

raise notice 'Cập nhật thành công!';

END;

$$ LANGUAGE plpgsql;

**Cập nhật thông tin khách:**

CREATE OR REPLACE FUNCTION update\_khach(old\_ho\_ten text,

old\_sdt text,

can\_ho text,

new\_ho\_ten text,

ngay\_sinh date,

sdt text,

email text,

thoi\_han\_dich\_vu text)

RETURNS void AS $$

declare ma\_cu\_dan int;

declare id int;

declare id int;

BEGIN

select "ID" from "Khách" into id where "Họ tên"=old\_ho\_ten and "SĐT"=old\_sdt;

if can\_ho is not null

then

update "Khách" set "Căn hộ"=can\_ho where "ID"=id;

end if;

if new\_ho\_ten is not null

then

update "Khách" set "Họ tên"=new\_ho\_ten where "ID"=id;

end if;

if ngay\_sinh is not null

then

update "Khách" set "Ngày sinh"=ngay\_sinh where "ID"=id;

end if;

if new\_sdt is not null

then

update "Khách" set "SĐT"=new\_sdt where "ID"=id;

end if;

if email is not null

then

update "Khách" set "Email"=email where "ID"=id;

end if;

if thoi\_han\_dich\_vu is not null

then

update "Khách" set "Thời hạn dịch vụ"=thoi\_han\_dich\_vu where "ID"=id;

end if;

END;

$$ LANGUAGE plpgsql;

**Tìm thông tin của admin:**

CREATE OR REPLACE FUNCTION info\_admin()

RETURNS TABLE( "Họ tên" character varying,

"Ngày sinh" date,

"SĐT" character varying,

"Email" character varying)

AS $$

BEGIN

return query

select "Cư dân"."Họ tên", "Cư dân"."Ngày sinh","Cư dân"."SĐT", "Cư dân"."Email"

from "Cư dân"

where "Cư dân"."Căn hộ"= 'admin';

END;

$$ LANGUAGE plpgsql;

**Tìm thông tin của căn hộ:**

CREATE OR REPLACE FUNCTION info\_can\_ho(room TEXT)

RETURNS TABLE("Căn hộ" character varying,

"Họ tên" character varying,

"SĐT" character varying,

"Email" character varying,

"Ngày sinh" date)

AS $$

declare

id int;

thoi\_gian date;

BEGIN

return query

select "Cư dân"."Căn hộ", "Cư dân"."Họ tên", "Cư dân"."SĐT", "Cư dân"."Email", "Cư dân"."Ngày sinh"

from "Cư dân"

where "Cư dân"."Căn hộ"= room

union all

select "Xe"."Căn hộ", "Biển số" as "Họ tên", "Loại xe" as "SĐT", "Trạng thái" as "Email", null as "Ngày sinh"

from "Xe"

where "Xe"."Căn hộ"= room;

END;

$$ LANGUAGE plpgsql;

**Tìm thông tin phí dịch vụ căn hộ:**

CREATE OR REPLACE FUNCTION info\_phi\_dv\_can\_ho(room TEXT)

RETURNS TABLE("Căn hộ" character varying,

"Thời gian" date,

"Số điện" integer,

"Số nước" integer,

"Phí vệ sinh" integer,

"Phí gửi xe" integer,

"Trạng thái" character varying)

AS $$

declare

id int;

thoi\_gian date;

BEGIN

return query

select "Phí dịch vụ"."Căn hộ", "Phí dịch vụ"."Thời gian", "Phí dịch vụ"."Số điện", "Phí dịch vụ"."Số nước", "Phí dịch vụ"."Phí vệ sinh", "Phí dịch vụ"."Phí gửi xe", "Phí dịch vụ"."Trạng thái"

from "Phí dịch vụ"

where public."Phí dịch vụ"."Căn hộ" = room;

END;

$$ LANGUAGE plpgsql;

select \* from info\_phi\_dv\_can\_ho('100');

**Tìm thông tin và lịch sử ra vào của xe:**

CREATE OR REPLACE FUNCTION info\_xe(bien\_so TEXT)

RETURNS TABLE(a character varying,

b character varying,

c character varying,

d timestamp with time zone,

e character varying) AS $$

declare

id int;

thoi\_gian date;

BEGIN

RETURN QUERY

SELECT "Xe"."Biển số" as a,

"Xe"."Căn hộ" as b,

"Xe"."Loại xe" as c,

null as d,

"Xe"."Trạng thái" as e

FROM "Xe"

where "Xe"."Biển số" = bien\_so

union all

SELECT "Lịch sử ra vào"."Biển số" as a,

null as b,

null as c,

"Lịch sử ra vào"."Thời gian" as d,

"Lịch sử ra vào"."In/Out" as e

FROM "Lịch sử ra vào"

where "Lịch sử ra vào"."Biển số" = bien\_so;

END;

$$ LANGUAGE plpgsql;

**Thêm phí dịch vụ hàng tháng:**

CREATE OR REPLACE FUNCTION phi\_dv(can\_ho TEXT,

so\_dien integer,

so\_nuoc integer)

RETURNS void AS $$

DECLARE

so\_nguoi int;

dien\_tich int;

phi\_do\_xe int;

ngay date;

BEGIN

select count("Cư dân"."Mã cư dân") into so\_nguoi from "Căn hộ"

join "Cư dân" on "Căn hộ"."Căn hộ" = "Cư dân"."Căn hộ"

where "Căn hộ"."Căn hộ" = can\_ho;

if (so\_nguoi = 0) then RAISE NOTICE 'Căn hộ không có người ở hoặc không tồn tại căn hộ %!', can\_ho; return; END IF;

select "Diện tích" into dien\_tich from "Căn hộ"

where "Căn hộ" = can\_ho;

select sum("Phí đỗ xe") into phi\_do\_xe from "Xe"

left join "Loại xe" on "Xe"."Loại xe"="Loại xe"."Loại xe"

where "Căn hộ" = can\_ho

group by "Xe"."Loại xe";

if phi\_do\_xe is null then select 0 into phi\_do\_xe; end if;

select CURRENT\_DATE into ngay;

INSERT INTO "Phí dịch vụ"("Căn hộ", "Thời gian", "Số điện", "Số nước", "Phí vệ sinh", "Phí gửi xe", "Trạng thái")

VALUES (can\_ho, ngay, so\_dien, so\_nuoc, 1500\*so\_nguoi\*dien\_tich, phi\_do\_xe, 'Chưa thanh toán');

RAISE NOTICE 'Thêm phí dịch vụ thành công cho căn hộ % ngày %!', can\_ho, ngay;

exception when unique\_violation then raise notice 'Thêm phí dịch vụ thành công cho căn hộ % ngày %!', can\_ho, ngay;

END;

$$ LANGUAGE plpgsql;

**Thanh toán phí dịch vụ:**

CREATE OR REPLACE FUNCTION thanh\_toan\_phi\_dv(can\_ho text, thang integer)

RETURNS void AS $$

declare key int;

BEGIN

select "PKey" into key from "Phí dịch vụ" where "Căn hộ"=can\_ho

and thang = (select EXTRACT(month FROM "Thời gian"))

and "Trạng thái"='Chưa thanh toán';

if can\_ho is null then raise notice 'Thanh toán không thành công!'; return; end if;

update "Phí dịch vụ" set "Trạng thái"='Đã thanh toán'

where "PKey"=key ;

raise notice 'Thanh toán thành công!';

END;

$$ LANGUAGE plpgsql;

**Cho xe vào hầm:**

CREATE OR REPLACE FUNCTION xe\_in(bien\_so text)

RETURNS void AS $$

DECLARE

trang\_thai text;

thoi\_gian TIMESTAMP;

can\_ho text;

BEGIN

select "Căn hộ" from "Xe" into can\_ho where "Biển số" = bien\_so;

if can\_ho is NULL then

select now() into thoi\_gian;

insert into "Xe"("Biển số", "Loại xe", "Trạng thái")

values (bien\_so, 'Xe vãng lai', 'In');

insert into "Lịch sử ra vào"("Biển số", "Thời gian", "In/Out")

values (bien\_so, thoi\_gian, 'In');

raise notice 'Đã cho xe % vào hầm!', bien\_so;

else

select now() into thoi\_gian;

update "Xe" set "Trạng thái" = 'IN' where "Biển số" = bien\_so;

insert into "Lịch sử ra vào"("Biển số", "Thời gian", "In/Out")

values (bien\_so, thoi\_gian, 'In');

raise notice 'Đã cho xe % vào hầm!', bien\_so;

end if;

exception WHEN unique\_violation THEN raise notice 'Xe % đã có trong hầm!', bien\_so;

END;

$$ LANGUAGE plpgsql;

**Cho xe ra khỏi hầm:**

CREATE OR REPLACE FUNCTION xe\_out(bien\_so text)

RETURNS void AS $$

DECLARE

bien\_so1 TEXT;

trang\_thai text;

thoi\_gian TIMESTAMP;

can\_ho text;

BEGIN

select "Biển số" from "Xe" into bien\_so1 where "Biển số" = bien\_so;

if bien\_so1 is null then raise notice 'Xe % không có trong hầm!', bien\_so; return;

end if;

select "Căn hộ" from "Xe" into can\_ho where "Biển số" = bien\_so;

if can\_ho is NULL then

select now() into thoi\_gian;

insert into "Lịch sử ra vào"("Biển số", "Thời gian", "In/Out")

values (bien\_so, thoi\_gian, 'Out');

perform delete\_xe(bien\_so);

raise notice 'Đã cho xe % ra khỏi hầm!', bien\_so;

else

select now() into thoi\_gian;

insert into "Lịch sử ra vào"("Biển số", "Thời gian", "In/Out")

values (bien\_so, thoi\_gian, 'Out');

update "Xe" set "Trạng thái" = 'Out' where "Biển số" = bien\_so;

raise notice 'Đã cho xe % ra khỏi hầm!', bien\_so;

end if;

END;  
$$ LANGUAGE plpgsql;

**IV.** **Đánh giá:**

**Cập nhật thông tin cư dân:**

- Cách 1:

CREATE OR REPLACE FUNCTION update\_cu\_dan(old\_ho\_ten text,

old\_room text,

new\_ho\_ten text,

new\_can\_ho text,

ngay\_sinh date,

sdt text,

email text)

RETURNS void AS $$

declare ma\_cu\_dan int;

BEGIN

select "Mã cư dân" from "Cư dân" into ma\_cu\_dan

WHERE "Họ tên"=old\_ho\_ten and "Căn hộ"=room ;

if new\_ho\_ten is not null

then

update "Cư dân" set "Họ tên"=new\_ho\_ten where "Mã cư dân"=ma\_cu\_dan;

end if;

if new\_can\_ho is not null

then

update "Cư dân" set "Căn hộ"=new\_can\_ho where "Mã cư dân"=ma\_cu\_dan;

end if;

if ngay\_sinh is not null

then

update "Cư dân" set "Ngày sinh"=ngay\_sinh where "Mã cư dân"=ma\_cu\_dan;

end if;

if sdt is not null

then

update "Cư dân" set "SĐT"=sdt where "Mã cư dân"=ma\_cu\_dan;

end if;

if email is not null

then

update "Cư dân" set "Email"=email where "Mã cư dân"=ma\_cu\_dan;

end if;

END;

$$ LANGUAGE plpgsql;

- Cách 2: Viết từng truy vấn cập nhập trong từng trường thông tin. Ví dụ:

update "Cư dân"

set "Họ tên"=new\_ho\_ten

where "Họ tên"=old\_ho\_ten and "Căn hộ"=room;

=> Nếu muốn update nhiều hơn 1 trường thông tin thì sẽ phải viết truy vấn nhiều lần

=> cách 1 sẽ tối ưu hơn

**Cập nhật thông tin xe:**

- Cách 1:

CREATE OR REPLACE FUNCTION update\_xe(old\_bien\_xe text,

new\_bien\_xe text,

can\_ho text,

loai\_xe text)

RETURNS void AS $$

declare ma\_cu\_dan int;

BEGIN

if new\_bien\_xe is not null

then

update "Xe" set "Biển số"=new\_bien\_xe

where "Biển số"=old\_bien\_xe;

end if;

if can\_ho is not null

then

update "Xe" set "Căn hộ"=can\_ho

where "Biển số"=old\_bien\_xe;

end if;

if loai\_xe is not null

then

update "Xe" set "Loại xe"=loai\_xe

where "Biển số"=old\_bien\_xe;

end if;

END;

$$ LANGUAGE plpgsql;

- Cách 2: Viết từng truy vấn cập nhập trong từng trường thông tin. Ví dụ:

update "Xe" set "Biển số"=new\_bien\_xe where "Biển số"=old\_bien\_xe;end if;

=> Nếu muốn update nhiều hơn 1 trường thông tin thì sẽ phải viết truy vấn nhiều lần

=> cách 1 sẽ tối ưu hơn

**Cập nhật thông tin phí dịch vụ:**

- Cách 1:

CREATE OR REPLACE FUNCTION update\_phi\_dv(can\_ho text,

thang integer,

so\_dien integer,

so\_nuoc integer,

phi\_vs integer,

phi\_xe integer)

RETURNS void AS $$

declare ma\_cu\_dan int;

declare id int;

BEGIN

select "PKey" into id from "Phí dịch vụ"

where "Căn hộ"=can\_ho

and thang = (select EXTRACT(month FROM "Thời gian"))

and (select EXTRACT(year FROM "Thời gian"))=(select EXTRACT(year FROM now()));

if id is null then raise notice 'Không có thông tin!'; return; end if;

if so\_dien is not null

then

update "Phí dịch vụ" set "Số điện"=so\_dien where "PKey"=id;

end if;

if so\_nuoc is not null

then

update "Phí dịch vụ" set "Số nước"=so\_nuoc where "PKey"=id;

end if;

if phi\_vs is not null

then

update "Phí dịch vụ" set "Phí vệ sinh"=phi\_vs where "PKey"=id;

end if;

if phi\_xe is not null

then

update "Phí dịch vụ" set "Phí gửi xe"=phi\_xe where "PKey"=id;

end if;

raise notice 'Cập nhật thành công!';

END;

$$ LANGUAGE plpgsql;

- Cách 2:

update "Phí dịch vụ"

set "Phí vệ sinh"=phi\_vs

where where "Căn hộ"=can\_ho

and thang = (select EXTRACT(month FROM "Thời gian"))

and (select EXTRACT(year FROM "Thời gian"))=(select EXTRACT(year FROM now()));

=> Nếu muốn update nhiều hơn 1 trường thông tin thì sẽ phải viết truy vấn nhiều lần

=> cách 1 sẽ tối ưu hơn

**Cập nhật thông tin khách:**

- Cách 1:

CREATE OR REPLACE FUNCTION update\_khach(old\_ho\_ten text,

old\_sdt text,

can\_ho text,

new\_ho\_ten text,

ngay\_sinh date,

sdt text,

email text,

thoi\_han\_dich\_vu text)

RETURNS void AS $$

declare ma\_cu\_dan int;

declare id int;

declare id int;

BEGIN

select "ID" from "Khách" into id where "Họ tên"=old\_ho\_ten and "SĐT"=old\_sdt;

if can\_ho is not null

then

update "Khách" set "Căn hộ"=can\_ho where "ID"=id;

end if;

if new\_ho\_ten is not null

then

update "Khách" set "Họ tên"=new\_ho\_ten where "ID"=id;

end if;

if ngay\_sinh is not null

then

update "Khách" set "Ngày sinh"=ngay\_sinh where "ID"=id;

end if;

if new\_sdt is not null

then

update "Khách" set "SĐT"=new\_sdt where "ID"=id;

end if;

if email is not null

then

update "Khách" set "Email"=email where "ID"=id;

end if;

if thoi\_han\_dich\_vu is not null

then

update "Khách" set "Thời hạn dịch vụ"=thoi\_han\_dich\_vu where "ID"=id;

end if;

END;

$$ LANGUAGE plpgsql;

- Cách 2:

update "Khách"

set "Họ tên"=new\_ho\_ten

where "Họ tên"=old\_ho\_ten and "SĐT"=old\_sdt;

=> Nếu muốn update nhiều hơn 1 trường thông tin thì sẽ phải viết truy vấn nhiều lần

=> cách 1 sẽ tối ưu hơn

**Tìm thông tin của căn hộ:**

- Cách 1:

CREATE OR REPLACE FUNCTION info\_can\_ho(room TEXT)

RETURNS TABLE("Căn hộ" character varying,

"Họ tên" character varying,

"SĐT" character varying,

"Email" character varying,

"Ngày sinh" date)

AS $$

declare

id int;

thoi\_gian date;

BEGIN

return query

select "Cư dân"."Căn hộ", "Cư dân"."Họ tên", "Cư dân"."SĐT", "Cư dân"."Email", "Cư dân"."Ngày sinh"

from "Cư dân"

where "Cư dân"."Căn hộ"= room

union all

select "Xe"."Căn hộ", "Biển số" as "Họ tên", "Loại xe" as "SĐT", "Trạng thái" as "Email", null as "Ngày sinh"

from "Xe"

where "Xe"."Căn hộ"= room;

END;

$$ LANGUAGE plpgsql;

- Cách 2:

select \* from "Cư dân"

join "Xe" on "Cư dân"."Căn hộ"="Xe"."Căn hộ"

where "Cư dân"."Căn hộ"=can\_ho;

=> Truy vấn ở cách 2 sẽ trả về nhiều giá trị trùng nhau do thực hiện phép tích Đề-các giữa 2 bảng "Cư dân" và "Xe" nên cách 1 sẽ tối ưu hơn.

**Thêm phí dịch vụ hàng tháng:**

CREATE OR REPLACE FUNCTION phi\_dv(can\_ho TEXT,

so\_dien integer,

so\_nuoc integer)

RETURNS void AS $$

DECLARE

so\_nguoi int;

dien\_tich int;

phi\_do\_xe int;

ngay date;

BEGIN

select count("Cư dân"."Mã cư dân") into so\_nguoi from "Căn hộ"

join "Cư dân" on "Căn hộ"."Căn hộ" = "Cư dân"."Căn hộ"

where "Căn hộ"."Căn hộ" = can\_ho;

if (so\_nguoi = 0) then RAISE NOTICE 'Căn hộ không có người ở hoặc không tồn tại căn hộ %!', can\_ho; return; END IF;

select "Diện tích" into dien\_tich from "Căn hộ"

where "Căn hộ" = can\_ho;

select sum("Phí đỗ xe") into phi\_do\_xe from "Xe"

left join "Loại xe" on "Xe"."Loại xe"="Loại xe"."Loại xe"

where "Căn hộ" = can\_ho

group by "Xe"."Loại xe";

if phi\_do\_xe is null then select 0 into phi\_do\_xe; end if;

select CURRENT\_DATE into ngay;

INSERT INTO "Phí dịch vụ"("Căn hộ", "Thời gian", "Số điện", "Số nước", "Phí vệ sinh", "Phí gửi xe", "Trạng thái")

VALUES (can\_ho, ngay, so\_dien, so\_nuoc, 1500\*so\_nguoi\*dien\_tich, phi\_do\_xe, 'Chưa thanh toán');

RAISE NOTICE 'Thêm phí dịch vụ thành công cho căn hộ % ngày %!', can\_ho, ngay;

exception when unique\_violation then raise notice 'Thêm phí dịch vụ thành công cho căn hộ % ngày %!', can\_ho, ngay;

END;

$$ LANGUAGE plpgsql;

=> Có thể gọi hàm sau để tự động thêm phí dịch vụ cho tất cả căn hộ nhưng vẫn phải update lại số điện, nước:

CREATE OR REPLACE FUNCTION process\_all\_phi\_dv()

RETURNS void AS $$

DECLARE

can\_ho text;

can\_ho\_cursor CURSOR FOR

SELECT "Căn hộ"

FROM "Căn hộ";

BEGIN

OPEN can\_ho\_cursor;

LOOP

FETCH can\_ho\_cursor INTO can\_ho;

EXIT WHEN NOT FOUND;

-- Gọi hàm xử lý từng căn hộ

PERFORM phi\_dv(can\_ho, 1, 1);

END LOOP;

CLOSE can\_ho\_cursor;

END;

$$ LANGUAGE plpgsql;

=> Query complete 00:00:04.404 ~ 00:00:04.919

**Thanh toán phí dịch vụ:**

CREATE OR REPLACE FUNCTION thanh\_toan\_phi\_dv(can\_ho text, thang integer)

RETURNS void AS $$

BEGIN

select "Căn hộ" into can\_ho from "Phí dịch vụ" where "Căn hộ"=can\_ho

and thang = (select EXTRACT(month FROM "Thời gian"))

and "Trạng thái"='Chưa thanh toán';

if can\_ho is null

then

raise notice 'Thanh toán không thành công!';

return;

end if;

update "Phí dịch vụ" set "Trạng thái"='Đã thanh toán'

where "Căn hộ"=can\_ho

and thang = (select EXTRACT(month FROM "Thời gian"))

and "Trạng thái"='Chưa thanh toán';

raise notice 'Thanh toán thành công!';

END;

$$ LANGUAGE plpgsql;

=> Query complete 00:00:00.077 ~ 00:00:00.163

- Cách 2:

CREATE OR REPLACE FUNCTION thanh\_toan\_phi\_dv(can\_ho text, thang integer)

RETURNS void AS $$

declare key int;

BEGIN

select "PKey" into key from "Phí dịch vụ" where "Căn hộ"=can\_ho

and thang = (select EXTRACT(month FROM "Thời gian"))

and "Trạng thái"='Chưa thanh toán';

if can\_ho is null then raise notice 'Thanh toán không thành công!';

return;

end if;

update "Phí dịch vụ" set "Trạng thái"='Đã thanh toán'

where "PKey"=key ;

raise notice 'Thanh toán thành công!';

END;

$$ LANGUAGE plpgsql;

=> Query complete 00:00:00.062 ~ 00:00:00.172

=> Cách 1 truy vấn UPDATE sẽ mất thời gian hơn cách 2 do cách 2 tìm bản ghi theo duy nhất khóa chính => thực hiện nhanh hơn.

**V.** **Kết luận:**

Sau quá trình thực hiện đồ án Database Hệ thống quản lý dữ liệu dân cư và dịch vụ khu chung cư, nhóm chúng em thu hoạch được:

* Kỹ năng thuyết trình, trình bày dự án.
* Phân công công việc và làm việc nhóm trong một dự án.
* Hiểu quy trình thiết kế cơ sở dữ liệu từ quá trình tìm hiểu đề tài, mô tả bài toán tới thiết kế mô hình sơ đồ liên kết thực thể, sơ đồ quan hệ.
* Hiểu cách kiểm tra, tính toán hiệu năng, tối ưu hóa truy vấn của hệ thống cơ sở dữ liệu PostgreSQL.
* Thu hoạch sản phẩm hệ thống quản lý dữ liệu dân cư và dịch vụ khu chung cư.