

MỤC LỤC

MỤC LỤC.....	1
I. ĐẶC TẢ HỆ THỐNG.....	2
1. Phân tích bài toán.....	2
2. Các chức năng.....	3
3. Công nghệ sử dụng.....	3
4. Phân chia công việc.....	3
II. Xây dựng cơ sở dữ liệu.....	4
1. Bảng customer.....	4
2. Bảng package.....	5
3. Bảng paymethod.....	5
4. Bảng address.....	5
5. Bảng transaction.....	6
6. Bảng bill.....	7
7. Bảng pay_history.....	7
8. Sơ đồ thực thể - liên kết.....	7
III. Các câu truy vấn.....	10
1. Trần Bảo Ngọc.....	10
2. Phạm Công Minh.....	14
3. Hà Văn Nam.....	18
IV. Các chức năng đã thực hiện.....	20
V. Những khó khăn khi thực hiện bài toán.....	23

I. ĐẶC TẢ HỆ THỐNG

Bài toán: Cơ sở dữ liệu cho nhà cung cấp dịch vụ Internet (ISP)

1. Phân tích bài toán

Hiện nay, nhu cầu sử dụng Internet của người dân Việt Nam rất cao và vẫn đang tăng. Vậy nên ý tưởng của nhóm là đứng vào vị trí của một nhà cung cấp dịch vụ Internet (ISP) xây dựng một trang web giúp khách hàng có thể đăng ký sử dụng mạng một cách đơn giản nhất.

- ❖ Để sử dụng dịch vụ Internet, đầu tiên khách hàng cần phải đăng ký một tài khoản trên trang web. Sau khi có tài khoản, khách hàng có thể tra cứu thông tin cá nhân đã điền trước đó (có thể xem luôn thông tin gói mạng đang sử dụng nếu đã đăng ký sử dụng mạng).
- ❖ Khách hàng có thể tra cứu thông tin dịch vụ có sẵn tại khu vực mình ở (có những tỉnh không có dịch vụ), nhập một đoạn ký tự và hệ thống sẽ gợi ý những thông tin liên quan về gói mạng dựa vào độ yêu thích và số lượng người sử dụng.
- ❖ Khách hàng chọn gói mạng muốn đăng ký, số tháng muốn đăng ký và phương thức thanh toán rồi sau đó thanh toán để hoàn tất quá trình đăng ký. Trong trường hợp khách hàng đang sử dụng dịch vụ thì sẽ có thông báo là có muốn đăng ký mới hay không. Nếu tiếp tục đăng ký thì khách hàng chọn số tháng đăng ký thêm và thanh toán để hoàn tất quá trình.
- ❖ Khi muốn gia hạn gói, nếu người dùng chưa đăng ký thì không thể gia hạn gói, nếu đã đăng ký thì có hai khả năng:
 - Nếu gói chưa hết hạn thì sẽ được gia hạn đến ngày được tính theo công thức:
Ngày hết hạn mới = Ngày hết hạn cũ + Số ngày đăng ký thêm
 - Nếu gói đã hết hạn thì:
Ngày hết hạn mới = Ngày đăng ký + Số ngày đăng ký thêm
- ❖ Sau khi đăng nhập vào hệ thống, khách hàng có thể tra cứu lịch sử giao dịch, hệ thống sẽ trả về số tiền đã giao dịch, chi tiết các hóa đơn để khách hàng dễ dàng kiểm tra.
- ❖ Khi muốn hủy gói, hệ thống hỏi để xác nhận lại, nếu khách hàng xác nhận thì hệ thống sẽ tiến hành hủy gói cước theo yêu cầu. Khi kiểm tra lại thông tin tài khoản sẽ có thông báo: “Bạn chưa đăng ký gói”.
- ❖ Trong quá trình đăng ký và gia hạn, hệ thống sẽ hỏi đánh giá về gói mạng tương ứng từ khách hàng. Điểm đánh giá sẽ được lưu trực tiếp vào hệ thống và được tính bởi công thức:

$$\text{new_rate} = \frac{\text{old_rate} \times \text{num_users} + \text{transaction_rate}}{\text{num_user} + 1}$$

2. Các chức năng

Từ những phân tích về bài toán ở trên thì nhóm quyết định triển khai các chức năng sau cho trang web:

- Đăng ký tài khoản
- Tra cứu thông tin tài khoản
- Tra cứu thông tin các gói mạng
- Đăng ký gói mạng mới
- Thanh toán hóa đơn, gia hạn gói
- Tra cứu lịch sử giao dịch
- Hủy gói, đổi gói
- Đánh giá về dịch vụ

3. Công nghệ sử dụng

- ❖ Front-end: HTML, CSS, Python, Bootstrap
- ❖ Back-end: Python Django
- ❖ Cơ sở dữ liệu: PostgreSQL

4. Phân chia công việc

a) Trần Bảo Ngọc

- Thiết kế cơ sở dữ liệu
- Thiết kế, xây dựng web
- 10 câu truy vấn

b) Phạm Công Minh

- Thiết kế cơ sở dữ liệu
- 10 câu truy vấn
- Viết báo cáo

c) Hà Văn Nam

- Nhập dữ liệu vào cơ sở dữ liệu
- 10 câu truy vấn
- Làm slide báo cáo

II. Xây dựng cơ sở dữ liệu

1. Bảng customer

Mục đích: Lưu trữ thông tin của khách hàng đăng ký và sử dụng dịch vụ.

Gồm có các thuộc tính:

STT	Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Ghi chú
1	username	Tên đăng nhập của khách hàng	varchar	Khóa chính
2	package_id	ID gói mạng khách hàng sử dụng	int	Khóa ngoài tham chiếu đến package(package_id)
3	city	Thành phố	char[]	Khóa ngoài tham chiếu đến address(address)
4	password	Mật khẩu	varchar	
5	first_name	Tên khách hàng	varchar	
6	last_name	Họ và tên đệm	varchar	
7	dob	Ngày sinh	date	
8	phone	Số điện thoại	varchar	
9	register_day	Ngày đăng ký	date	
10	expire_day	Ngày hết hạn	date	
11	address	Địa chỉ	varchar	
12	customer_id	ID khách hàng	int	
13	cmt	Số CMT/CCCD	varchar	
14	gender	Giới tính	char	

2. Bảng package

Mục đích: Lưu trữ thông tin các gói mạng mà nhà mạng cung cấp.

Gồm có các thuộc tính:

STT	Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Ghi chú
1	package_id	ID gói mạng	int	Khóa chính
2	package_name	Tên gói mạng	varchar	
3	cost	Giá tiền	int	
4	speed	Tốc độ đường truyền	varchar	
5	number_user	Số lượng người dùng	int	
6	rate	Đánh giá từ người dùng	real	

3. Bảng paymethod

Mục đích: Lưu trữ thông tin các phương thức thanh toán

Gồm có các thuộc tính:

STT	Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Ghi chú
1	pay_method	Tên phương thức	varchar	Khóa chính
2	discription	Mô tả phương thức	varchar	

4. Bảng address

Mục đích: Lưu trữ thông tin các thành phố được cung cấp dịch vụ.

Gồm có các thuộc tính:

STT	Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Ghi chú
1	address	Tên thành phố	varchar	Khóa chính
2	package_id	ID gói mạng có sẵn	varchar	Khóa chính Khóa ngoài tham chiếu đến package(package_id)

5. Bảng transaction

Mục đích: Lưu trữ thông tin các giao dịch.

Gồm có các thuộc tính:

STT	Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Ghi chú
1	tran_id	ID giao dịch	int	Khóa chính
2	package_id	ID gói mạng khách hàng đăng ký	int	Khóa ngoài tham chiếu đến package(package_id)
3	username	Khách hàng thực hiện giao dịch	varchar	Khóa ngoài tham chiếu đến customer(username)
4	bill_id	ID hóa đơn	int	
5	total	Tổng số tiền	int	
6	register_day	Ngày giao dịch	date	
7	expire_day	Ngày hết hạn	date	
8	rate	Đánh giá từ khách hàng	real	

6. Bảng bill

Mục đích: Lưu trữ thông tin hóa đơn.

Gồm có các thuộc tính:

STT	Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Ghi chú
1	bill_id	ID hóa đơn	int	Khóa chính
2	username	Khách hàng	varchar	Khóa ngoài tham chiếu đến customer(username)
3	pay_method	Tên phương thức thanh toán	varchar	Khóa ngoài tham chiếu đến paymethod(pay_method)
4	total	Tổng số tiền	numeric	
5	register_day	Ngày đăng ký	date	

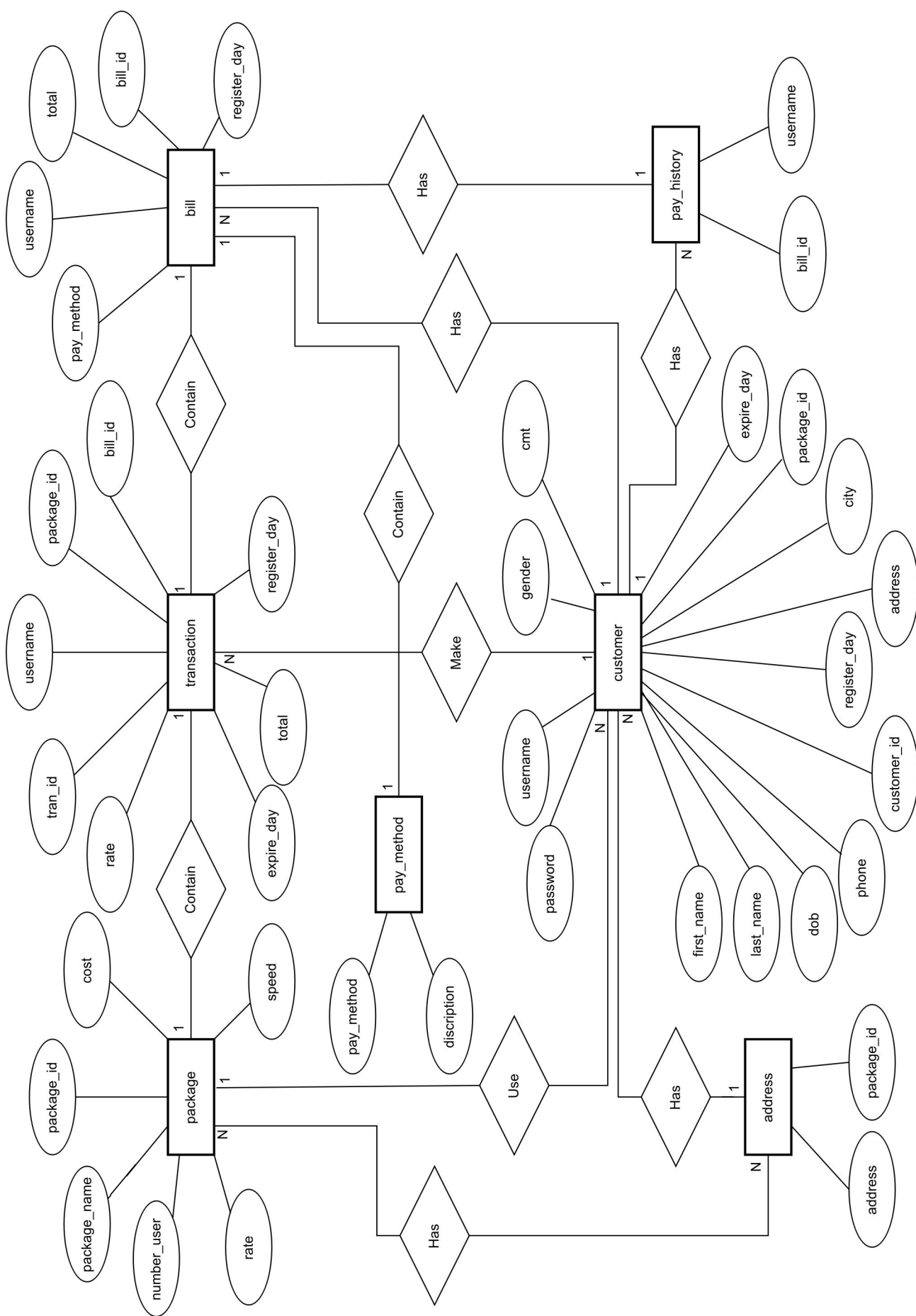
7. Bảng pay_history

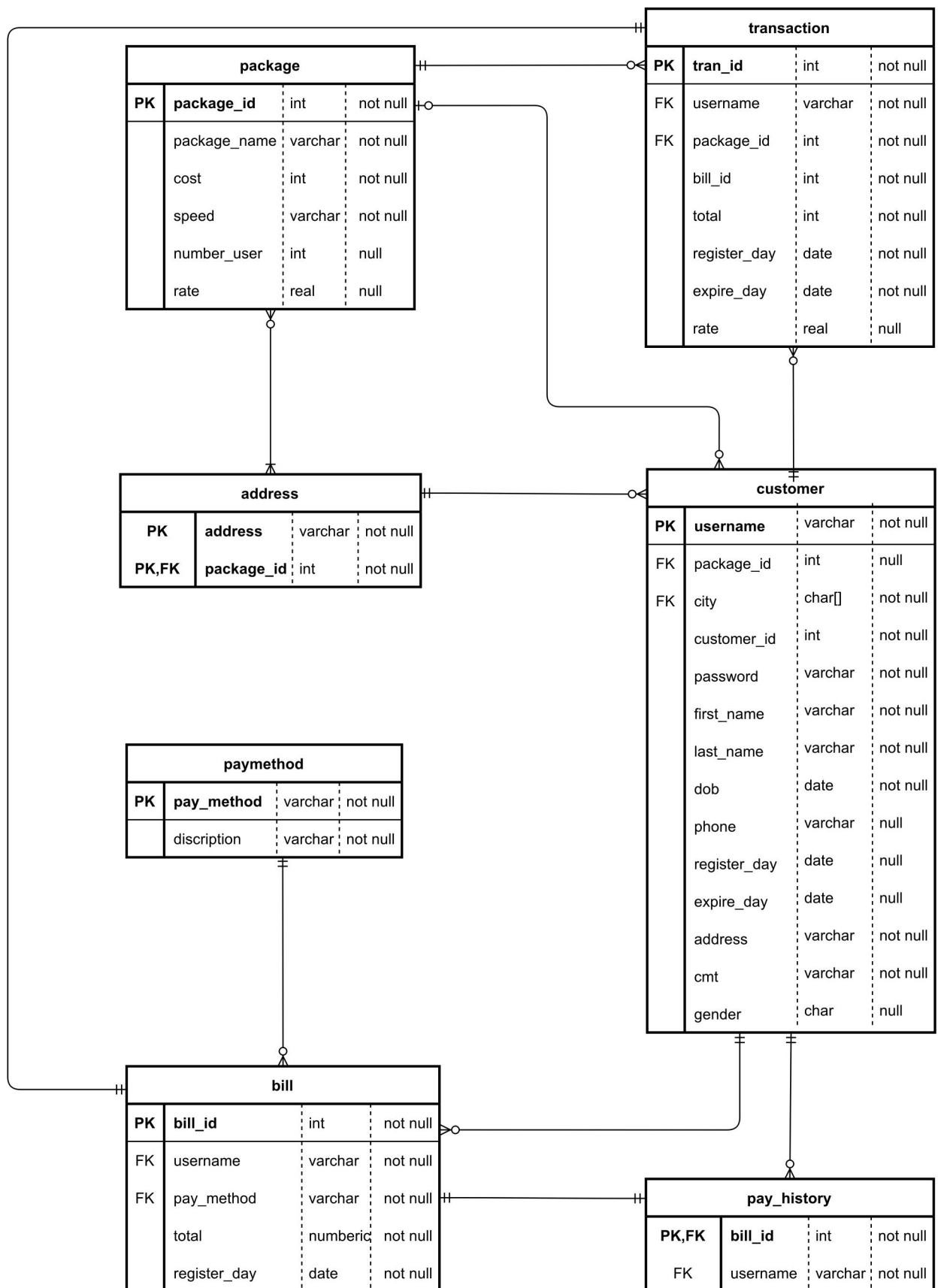
Mục đích: Lưu trữ lịch sử giao dịch.

Gồm có các thuộc tính:

STT	Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Ghi chú
1	bill_id	ID hóa đơn	int	Khóa chính Khóa ngoài tham chiếu đến bill(bill_id)
2	username	Khách hàng	varchar	Khóa ngoài tham chiếu đến customer(username)

8. Sơ đồ thực thể - liên kết





III. Các câu truy vấn

1. Trần Bảo Ngọc

1.1. Tạo trigger để khi thêm mỗi giao dịch, các thông tin tự động update trên các bảng để đảm bảo thông tin không bị sai lệch.

Khi thêm 1 giao dịch thì cần phải có những thay đổi sau:

- Đối với bảng customer: Cập nhật ngày hết hạn, gói đăng ký.
- Đối với bảng bill: Thêm 1 bill mới và các thông tin của bill.
- Đối với bảng package: Số người sử dụng = số người sử dụng cũ + 1
- Đánh giá mới = (đánh giá cũ \times số người dùng cũ + đánh giá trên giao dịch) \div (số người sử dụng mới)

```
CREATE trigger af_insert_tran
After insert on transaction
for each row
Execute procedure insert_tran();
CREATE OR REPLACE FUNCTION insert_tran() RETURNS TRIGGER AS $$
BEGIN
Update customer
Set
package_id = New.package_id,
expire_day = New.expire_day
where username = new.username;
update package
set
rate = ((rate*number_user + new.rate)/(number_user+1)),
number_user = number_user + 1
where package_id = new.package_id;
insert into bill(bill_id,username,total,register_day)
values(new.bill_id,new.username,new.total,new.register_day);
return new;
END;
$$ LANGUAGE plpgsql;
```

1.2. Tạo trigger để khi thêm 1 bill mới thì lịch sử giao dịch của user thanh toán bill đó được update để tra cứu

```
create trigger af_insert_bill
after insert on bill
for each row
execute procedure insert_bill()
CREATE OR REPLACE FUNCTION insert_bill() RETURNS TRIGGER AS $$
BEGIN
insert into pay_history values (new.username,new.bill_id);
return new;
END;
$$ LANGUAGE plpgsql;
```

1.3. Lấy ra các gói hỗ trợ người dùng theo mục đích doanh nghiệp hoặc gia đình và in theo đánh giá từ cao tới thấp

```
select * from package
where lower(package_name) like '%net%'
order by speed;
```

❖ Chú thích

“net” là gói gia đình.

“fast” là gói doanh nghiệp.

Còn lại là gói đặc biệt.

1.4. Tính tổng doanh thu mỗi năm

```
select sum(total) as doanh_thu,count(*) as so_luong_hoa_don
from bill
where date_part('year', bill.register_day) = '2020';
```

1.5. Danh sách các khách hàng đã hết hạn gói

```
select * from customer
where expire_day < CURRENT_DATE;
```

1.6. Tổng số tiền đã giao dịch và số lượng giao dịch đã thực hiện của các user

```
select count(b.*),sum(b.total),c.*
from bill b,customer c
where b.username = c.username
group by c.username;
```

1.7. Top 10 tỉnh có doanh thu cao nhất năm

```
SELECT customer.city, SUM(transaction.total)
FROM transaction JOIN customer USING (username)
WHERE date_part('year', transaction.register_day) = 2020
GROUP BY customer.city
ORDER BY "Tổng doanh thu" DESC
LIMIT 10;
```

1.8. Đưa ra những người dùng hoạt động nhiều nhất

- ❖ Chú thích: Là người dùng có trên nhiều giao dịch nhất và tổng số tiền đã thanh toán phải nhiều hơn 3 triệu.

```
WITH temp AS
(SELECT username, SUM(total) sum, count(*) tran
FROM transaction
GROUP BY username)
SELECT c.username, temp.sum AS "Tổng tiền" ,temp.tran as "Số giao dịch"
FROM temp,customer c
WHERE
temp.username = c.username
and
temp.tran = (SELECT MAX(tran) FROM temp)
And temp.sum > 3000000
```

1.9. Đưa ra danh sách những người dùng (username, phone) sử dụng các gói có giá trị từ 1 triệu trở lên mà 2 ngày nữa là hết hạn để gửi tin nhắn thông báo gia hạn kịp thời

```
select c.username,c.phone
from customer c,package p
where c.package_id = p.package_id
and c.expire_day = CURRENT_DATE - integer '2'
and p.cost >= 1000000
```

1.10. Đưa ra thống kê về số người sử dụng của mỗi gói theo các thành phố khác nhau và số người hiện còn đang sử dụng gói đó

```
with temp as(select count(t.*) num,c.city city,p.package_id pname
from transaction t,customer c,address a,package p
where t.username = c.username
and c.city = a.address
and p.package_id = a.package_id
and c.package_id = t.package_id
and c.package_id = p.package_id
and c.expire_day >= CURRENT_DATE
group by c.city,p.package_id
)
select count(t.*) as "Số người sử dụng",a.address,p.package_name,temp.num
as "Số người đang sử dụng"
from address a,package p,transaction t,temp
where a.package_id = p.package_id
and t.package_id = p.package_id
and temp.city = a.address
and temp.pname = p.package_id
group by a.address,p.package_name,temp.num
```

2. Phạm Công Minh

2.1. Tìm thông tin những khách hàng ở một địa chỉ nhất định có sinh nhật vào hôm nay

```
SELECT customer.* FROM customer
WHERE city = 'Nam Định'
AND date_part('day', customer.dob) = date_part('day', CURRENT_DATE)
AND date_part('month', customer.dob) = date_part('month', CURRENT_DATE);
```

2.2. Chọn địa chỉ, chọn khoảng giá và list ra các gói có sẵn

```
SELECT package_name, speed, cost::money, number_user,
       ROUND(rate::numeric,2) "Rate"
FROM address LEFT JOIN package
ON address.package_id = package.package_id
WHERE address = 'Hà Nội'
AND cost BETWEEN 150000 AND 250000
ORDER BY "cost" DESC;
```

2.3. Độ phổ biến của các cách thức thanh toán (số lượt sử dụng, doanh thu) ở thành phố bất kỳ

```
WITH temp_method AS
(SELECT customer.city,
       bill.pay_method "Phương thức",
       COUNT(bill.bill_id) "Lượt sử dụng",
       SUM(bill.total)::money "Doanh thu"
FROM bill JOIN customer USING (username)
GROUP BY bill.pay_method, customer.city)
SELECT city, "Phương thức", "Lượt sử dụng", "Doanh thu"
FROM temp_method
WHERE city = 'Quảng Ninh'
```

2.4. Khách hàng đã thanh toán nhiều tiền nhất

```
WITH temp_total AS
(SELECT username, SUM(total) temp_sum FROM bill
GROUP BY username)
SELECT username, customer.first_name, customer.last_name, customer.phone,
       temp_total.temp_sum::money AS "Số tiền"
FROM temp_total JOIN customer USING (username)
WHERE temp_total.temp_sum = (
    SELECT MAX(temp_sum) FROM temp_total
);
```

2.5. Doanh thu tháng (tùy chọn) của tỉnh

```
SELECT customer.city, SUM(total)::money
FROM transaction JOIN customer USING (username)
WHERE date_part('year', transaction.register_day) = 2020
AND date_part('month', transaction.register_day) = 11
AND customer.city = 'Nam Định'
GROUP BY customer.city;
```

2.6. Những khách hàng tháng này có gói hết hạn

```
SELECT username, customer.city, customer.first_name, customer.last_name,
       transaction.expire_day
FROM transaction JOIN customer USING (username)
WHERE date_part('year', CURRENT_DATE) = date_part('year',
       transaction.expire_day)
AND date_part('month', CURRENT_DATE) = date_part('month',
       transaction.expire_day);
```

2.7. Đếm số giao dịch trong tháng của thành phố/tỉnh

```
SELECT customer.city, COUNT(bill.bill_id) AS "Số giao dịch"
FROM bill JOIN customer USING (username)
WHERE date_part('year', bill.register_day) = date_part('year', CURRENT_DATE)
AND date_part('month', bill.register_day) = date_part('month', CURRENT_DATE)
GROUP BY customer.city
ORDER BY "Số giao dịch" DESC;
```

2.8. Điểm rate trung bình theo tháng của các tỉnh

```
WITH temp_rate AS
(SELECT customer.city, date_part('month', transaction.register_day) "Tháng",
        date_part('year', transaction.register_day) "Năm",
        ROUND(AVG(transaction.rate), 2) "Rate trung bình"
FROM transaction JOIN customer USING (username)
GROUP BY "Tháng", "Năm", customer.city
)
SELECT city, "Tháng", "Năm", "Rate trung bình"
FROM temp_rate
WHERE "Tháng" = 12 AND "Năm" = 2020
ORDER BY "Rate trung bình" DESC;
```

2.9. Tăng trưởng doanh thu theo từng tháng (của toàn hệ thống)

- ❖ Công thức tính tăng trưởng doanh thu:

$$\text{growth_rate} = \frac{\text{last_value} - \text{first_value}}{\text{first_value}} \times 100$$

```
WITH avg_revenue AS
(SELECT date_trunc('month', register_day)::date AS m,
        SUM(total) revenue
FROM transaction
GROUP BY m
)
SELECT date_part('year', m) "Năm",
        date_part('month', m) "Tháng",
        revenue::money "Doanh thu",
        ROUND((((revenue - lag(revenue, 1) OVER (ORDER BY m ASC)) / ((lag(revenue, 1)
OVER (ORDER BY m ASC))::real)*100)::numeric, 3) "Tăng trưởng (%)"
FROM avg_revenue
```


2.10. Tăng trưởng doanh thu theo từng tháng của thành phố/tỉnh

```
WITH avg_revenue AS
(
  SELECT customer.city,
         date_trunc('month', transaction.register_day)::date AS m,
         SUM(transaction.total) revenue
  FROM transaction JOIN customer USING (username)
  GROUP BY m, customer.city
)
SELECT city,
       date_part('year', m) "Năm",
       date_part('month', m) "Tháng",
       revenue::money "Doanh thu",
       ROUND(((revenue - lag(revenue, 1) OVER (ORDER BY m ASC)) / ((lag(revenue, 1)
OVER (ORDER BY m ASC))::real) * 100)::numeric, 3) "Tăng trưởng (%)"
FROM avg_revenue
WHERE city = 'Thái Bình'
```

3. Hà Văn Nam

3.1. Số hóa đơn trả bằng Momo

```
SELECT count(pay_method) AS Pay_Momo
FROM bill
where pay_method LIKE '%Momo';
```

3.2. Doanh thu của tháng 12 năm 2020 bất kì

```
SELECT sum(total) AS DoanhThuThang
FROM bill
WHERE register_day BETWEEN '2020-12-01' AND '2020-12-31';
```

3.3. Lấy ra số người dùng của các gói mạng theo số người dùng tăng dần

```
SELECT *
FROM package
ORDER BY number_user;
```

3.4. Tính rate trung bình của dịch vụ mạng

```
select AVG(rate) AS avgService
from package ;
```

3.5. Sắp xếp các bản ghi theo đánh giá từ cao đến thấp

```
SELECT *
FROM package
ORDER BY rate DESC;
```

3.6. Danh sách khách hàng tại tỉnh/thành phố, đăng kí trong tháng 11 năm 2020

```
SELECT first_name , last_name, register_day
FROM customer
WHERE address = 'Hải Phòng'
AND register_day BETWEEN '2020-11-01' and '2020-11-30';
```

3.7. Đếm số khách hàng tại tỉnh Bắc Ninh

```
SELECT COUNT(customer_id), address
FROM customer
WHERE address = 'Bắc Ninh'
GROUP BY address;
```

3.8. Lấy ra danh sách khách hàng , sử dụng gói mạng đắt nhất để chăm sóc đặc biệt

❖ Chú thích: Gói mạng đắt nhất có tên là “Best”

```
SELECT customer.first_name, customer.last_name, package.cost
FROM customer
LEFT JOIN package
ON customer.package_id = package.package_id
WHERE customer.package_id = package.package_id
AND package_name = 'Best';
```

3.9. Lấy ra thông tin khách hàng thanh toán bằng Zalo Pay , dùng gói mạng Best để nhận ưu đãi

```
select customer.first_name, customer.last_name , package.package_name
from customer
right join package
on customer.package_id = package.package_id
where package.package_name = 'Best';
```

3.10. Tỷ lệ khách hàng sử dụng phương thức thanh toán ‘Gửi tiền mặt’

```
WITH tmp AS
(SELECT COUNT(*) tien_mat
FROM bill
WHERE pay_method = 'Gửi tiền mặt'
)
SELECT tmp.tien_mat/(COUNT(bill.*)::real)*100
FROM tmp, bill
GROUP BY tmp.tien_mat;
```

IV. Các chức năng đã thực hiện

1. Chức năng đăng ký, đăng nhập

a) Chức năng đăng ký

```
INSERT INTO customer(customer_id,username,password,first_name,last_name,dob,
phone,register_day,address,city,cmt,gender)
VALUES ('%s','%s','%s','%s','%s','%s','%s','%s','%s','%s','%s','%s');
```

Khách hàng nhập thông tin trong form đăng ký và thông tin sẽ được nhập vào cơ sở dữ liệu bằng câu lệnh trên.

b) Chức năng đăng nhập

```
SELECT username,password FROM customer;
```

Lấy ra các username trong hệ thống để kiểm tra xem tài khoản đã tồn tại hay chưa trong chức năng đăng ký và kiểm tra xem tài khoản nhập có chính xác không trong đăng nhập. Lấy kèm mật khẩu để đăng nhập.

2. Chức năng tìm kiếm gói mạng

a) Tìm kiếm theo mục đích

```
SELECT package_name,speed,rate,cost
FROM package
WHERE lower(package_name) like '%net%';
```

Các gói theo mục đích đều có thành phần tên gọi để chỉ trong đó. Trong gói có chữ 'net' là gói hỗ trợ cho gia đình còn chữ 'fast' là hỗ trợ cho doanh nghiệp. những gói không có 2 chữ đó là gói đặc biệt.

```
SELECT package_name,speed,rate,cost
FROM package
WHERE lower(package_name) NOT LIKE '%net%'
AND lower(package_name) NOT LIKE '%fast%';
```

b) Tìm kiếm gói theo thành phố

```
SELECT package_name,speed,rate,cost
FROM package
WHERE package_id IN
(SELECT package_id
FROM address
WHERE address = '%s');
```

Lấy ra các gói theo thành phố được lựa chọn theo tham số truyền vào là 1 xâu ký tự tên thành phố. Thông tin được lấy ra từ bảng package và address, thỏa mãn điều kiện package_id nằm trong những id mà thành phố đó hỗ trợ.

c) Khi user nhập vào 1 chuỗi ký tự để tìm kiếm

```
SELECT package_name
FROM package
WHERE lower(package_name) LIKE '%%%s%%';
```

```
SELECT address
FROM address
WHERE lower(address) LIKE '%%%s%%'
GROUP BY address;
```

Do để phân biệt với %s truyền tham số nên phải sử dụng %%% để hiện dấu phần trăm trong câu truy vấn.

3. Lấy ra phương thức thanh toán để giao dịch

```
SELECT pay_method, pay_method
FROM paymethod;
```

4. Kiểm tra người dùng đã hết hạn gói hay chưa

```
SELECT package_id, expire_day
FROM customer
WHERE username = '%s';
```

Lấy ra tên gói và ngày hết hạn của khách hàng từ bảng customer với tham số truyền vào là username. Nếu gói đã hết hạn thì khi đăng ký mới hoặc gia hạn thì ngày hết hạn mới được tính theo công thức : ngày hết hạn cũ + số ngày đã đăng ký

5. Thêm transaction vào cơ sở dữ liệu

```
INSERT INTO transaction
VALUES ('%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s');
```

Thêm vào bảng transaction các tham số tương ứng được nhập trong form giao dịch đăng ký hoặc gia hạn

6. Thêm các bảng thông tin tương ứng sau mỗi giao dịch

Khi thêm 1 giao dịch thì cần phải có những thay đổi sau:

- Đối với bảng customer: Cập nhật ngày hết hạn, gói đăng ký.
- Đối với bảng bill: Thêm 1 bill mới và các thông tin của bill.
- Đối với bảng package: Số người sử dụng = số người sử dụng cũ + 1
- Đánh giá mới = (đánh giá cũ \times số người dùng cũ + đánh giá trên giao dịch) \div (số người sử dụng mới)

```
CREATE trigger af_insert_tran
AFTER INSERT ON transaction
for each row
Execute procedure insert_tran();
CREATE OR REPLACE FUNCTION insert_tran() RETURNS TRIGGER AS $$
BEGIN
UPDATE customer
SET
package_id = New.package_id,
expire_day = New.expire_day
WHERE username = new.username;
UPDATE package
SET
rate = ((rate*number_user + new.rate)/(number_user+1)),
number_user = number_user + 1
WHERE package_id = new.package_id;
INSERT INTO bill(bill_id,username,total,register_day)
VALUES(new.bill_id,new.username,new.total,new.register_day);
return new;
END;
$$ LANGUAGE plpgsql;
```

7. Xuất ra tổng số tiền khách hàng đã thanh toán

```
SELECT sum(total)
FROM bill
WHERE username = '%s';
```

8. Xuất ra thông tin các hóa đơn

```
SELECT register_day,total,pay_method
FROM bill
WHERE username = '%s';
```

V. Những khó khăn khi thực hiện bài toán

1. Vấn đề với số người sử dụng và đánh giá (rate) của các gói

Thông thường số người sử dụng sẽ là số người hiện đang sử dụng gói đó cho dù đã hết hạn hay chưa. Nhưng nếu như vậy sẽ rất khó khăn cho việc tính đánh giá. Đồng thời do trang web chỉ là demo nên rất khó để cho người sử dụng rồi mới vào đánh giá rate. Do đó nhóm xử lý vấn đề đó như sau:

- ❑ Số người sử dụng sẽ được tính bằng số giao dịch gia hạn hoặc đăng ký gói đó. Khi thêm 1 giao dịch thì số người sử dụng tăng lên 1. Giống như cách xử lý của các ứng dụng trên google play là lượt tải về. Dù người này đã từng tải nhưng khi người đó gỡ và tải lại thì số lượt tải vẫn tăng lên chứ không giữ nguyên. Đồng thời khi hủy gói cũng sẽ không trừ bớt đi số người sử dụng.
- ❑ Về đánh giá thì sẽ cho đánh giá trực tiếp trong các giao dịch, coi như người đó đã sử dụng và đánh giá. Nếu người đó không đánh giá thì giá trị đánh giá mặc định là 3.0.
- ❑ Để kết quả đánh giá được tính bình quân thì sau mỗi giao dịch kết quả đánh giá sẽ được cập nhật mới như sau:

$\text{Đánh giá mới} = (\text{Đánh giá cũ} \times \text{số người sử dụng cũ} + \text{đánh giá mới}) \div (\text{số người sử dụng mới})$

2. Vấn đề với việc đăng ký gói và gia hạn gói

- ❑ Do mỗi người dùng chỉ có thể đăng ký một gói mạng nên khi đăng ký gói phải đảm bảo rằng người đó đang không sử dụng gói nào (chưa đăng ký hoặc gói hiện tại đã hết hạn). Từ đó có những trường hợp gói đang sử dụng còn nhiều ngày mà phải hủy sẽ mất rất nhiều tiền mà vẫn không được hoàn lại.
- ❑ Còn khi gia hạn thì phải đảm bảo người đó đang có sử dụng gói dù hết hạn hay chưa. Khi gia hạn gói thì sẽ chia làm 2 khả năng gói hiện tại chưa hết hạn hoặc đã hết hạn.

Xử lý vấn đề:

- Đối với việc muốn đổi gói thì trước tiên phải hủy gói hiện tại dù đã hết hạn hay chưa và không có hoàn trả cho việc hủy gói.
- Đối với gia hạn, nếu gói đã hết hạn thì ngày hết hạn mới bằng ngày gia hạn cộng số ngày gia hạn. Còn gói chưa hết hạn thì ngày hết hạn mới bằng ngày hết hạn cũ cộng số ngày gia hạn.

3. Vấn đề với loại giao dịch

Trên thực tế thì giao dịch phải chia làm các loại giao dịch riêng biệt như hủy gói, đăng ký, gia hạn, đổi gói. Nhưng trong bài project này của nhóm thì chưa có thuộc tính đó trong giao dịch nên khi tra cứu khách hàng chỉ có thể thấy được giao dịch đó mình sử dụng gói nào đến ngày bao nhiêu, không biết những lần hủy giao dịch. Và do đã xây dựng database và input dữ liệu nên nhóm quyết định không sửa phần đó nữa.

TỔNG KẾT