

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO
THỰC HÀNH CƠ SỞ DỮ LIỆU

CHỦ ĐỀ:

QUẢN LÝ HÓA ĐƠN TIỀN ĐIỆN CHO
KHÁCH HÀNG

Giảng viên hướng dẫn: TS. Trần Văn Đặng

Nhóm thực hiện: Nhóm 5

Vũ Quốc Bảo	20225694	CNTT Việt Nhật 07 K67
Vương Quốc Huy	20225637	CNTT Việt Nhật 07 K67
Đỗ Trung Hiếu	20225623	CNTT Việt Nhật 07 K67

HÀ NỘI – 2024

Table of Contents

I. Mô tả chủ đề	3
1. Mục đích	3
1.1. Đảm bảo tính chính xác và minh bạch	3
1.2. Quản lí tài chính hiệu quả	3
1.3. Cải thiện dịch vụ khách hàng	3
1.4. Khuyến khích tiết kiệm năng lượng	3
1.5. Hỗ trợ quản lí hạ tầng điện lực	3
1.6. Ứng dụng công nghệ	4
2. Tình huống ứng dụng	4
2.1. Thuận tiện trong thanh toán và lưu trữ	4
2.2. Tiết kiệm chi phí và bảo vệ môi trường	4
2.3. Tăng cường trải nghiệm khách hàng	4
2.4. Đối phó với tình huống khẩn cấp	4
3. Chức năng nghiệp vụ	4
II. Cấu trúc dữ liệu	6
1. Sơ đồ thực thể liên kết	6
2. Dữ liệu, yêu cầu, mối quan hệ	8
3. Truy vấn SQL	12
3.1. Function và Trigger	12
3.2. View	14
3.3. Query	15
III. Kết quả	26
IV. Thành viên nhóm	30

I. Mô tả chủ đề

Tiền điện là một phần không thể thiếu trong cuộc sống hàng ngày, đóng vai trò quan trọng trong việc duy trì sự tiện nghi và thoải mái. Quản lý và tính toán tiền điện không chỉ đảm bảo việc sử dụng điện năng hiệu quả mà còn giúp tiết kiệm chi phí và bảo vệ môi trường. Đối với các nhà cung cấp dịch vụ, việc quản lý hóa đơn chính xác và hỗ trợ khách hàng kịp thời là yếu tố then chốt để duy trì lòng tin và sự hài lòng. Công nghệ hiện đại, từ hệ thống đo lường thông minh đến hóa đơn điện tử, đã và đang góp phần nâng cao hiệu quả và sự tiện lợi trong quản lý tiền điện, mang lại lợi ích thiết thực cho cả người sử dụng và nhà cung cấp. Trong bối cảnh nhu cầu sử dụng điện ngày càng tăng, việc quản lý tiền điện hiệu quả càng trở nên cần thiết và quan trọng, giúp đảm bảo một tương lai bền vững và phát triển.

1. Mục đích

1.1. Đảm bảo tính chính xác và minh bạch

- *Ghi nhận chính xác mức tiêu thụ điện:* Quản lý hóa đơn tiền điện giúp đảm bảo rằng mức tiêu thụ điện được ghi nhận chính xác, tránh sai sót và bất công bằng.
- *Minh bạch trong tính toán chi phí:* Hóa đơn cần phải rõ ràng, minh bạch để khách hàng hiểu rõ các khoản chi phí họ phải trả.

1.2. Quản lý tài chính hiệu quả

- *Dự báo và quản lý chi phí:* Quản lý hóa đơn giúp khách hàng dự báo và kiểm soát chi phí điện năng hàng tháng, từ đó lên kế hoạch tài chính hợp lý.
- *Đảm bảo thu nhập cho nhà cung cấp:* Giúp các công ty điện lực đảm bảo thu nhập ổn định và có thể tái đầu tư vào hạ tầng và dịch vụ.

1.3. Cải thiện dịch vụ khách hàng

- *Hỗ trợ khách hàng:* Giúp giải quyết các thắc mắc và khiếu nại của khách hàng liên quan đến hóa đơn và mức tiêu thụ điện.
- *Nâng cao trải nghiệm khách hàng:* Cung cấp các dịch vụ tiện ích như thông báo hóa đơn, nhắc nhở thanh toán và hỗ trợ trực tuyến.

1.4. Khuyến khích tiết kiệm năng lượng

- *Theo dõi và kiểm soát mức tiêu thụ:* Khách hàng có thể theo dõi mức tiêu thụ điện và điều chỉnh hành vi sử dụng để tiết kiệm năng lượng.
- *Khuyến khích sử dụng hiệu quả:* Thông tin từ hóa đơn giúp khách hàng nhận biết các thiết bị tiêu thụ nhiều điện và tìm cách sử dụng hiệu quả hơn.

1.5. Hỗ trợ quản lý hạ tầng điện lực:

- *Phân tích và dự báo nhu cầu:* Dữ liệu từ hóa đơn giúp các công ty điện lực phân tích và dự báo nhu cầu điện năng, từ đó lập kế hoạch phát triển hạ tầng.
- *Quản lý tải và điều hòa điện:* Giúp quản lý và điều hòa tải điện để tránh quá tải và cắt điện.

1.6. Ứng dụng công nghệ

- *Hóa đơn điện tử*: Giảm thiểu việc sử dụng giấy và nâng cao tính tiện lợi cho khách hàng.
- *Hệ thống đo lường thông minh*: Sử dụng công nghệ đo lường thông minh để cải thiện độ chính xác và giảm thiểu sai sót.

→ Quản lý hóa đơn tiền điện hiệu quả không chỉ mang lại lợi ích cho cả nhà cung cấp dịch vụ và khách hàng mà còn góp phần vào sự phát triển bền vững và bảo vệ môi trường.

2. Tình huống ứng dụng

2.1. Thuận tiện trong thanh toán và lưu trữ

- *Tình huống*: Một hộ gia đình nhận hóa đơn tiền điện hàng tháng.
- *Lợi ích*: Hóa đơn được gửi qua email hoặc ứng dụng di động, giúp khách hàng dễ dàng thanh toán trực tuyến mà không cần chờ đợi hóa đơn giấy. Hóa đơn điện tử cũng được lưu trữ trên hệ thống, dễ dàng tra cứu khi cần.

2.2. Tiết kiệm chi phí và bảo vệ môi trường

- *Tình huống*: Công ty cung cấp điện muốn giảm chi phí in ấn và gửi hóa đơn.
- *Lợi ích*: Hóa đơn điện tử giúp tiết kiệm chi phí giấy, mực in, và chi phí vận chuyển. Việc sử dụng hóa đơn điện tử cũng góp phần bảo vệ môi trường, giảm lượng rác thải từ giấy.

2.3. Tăng cường trải nghiệm khách hàng

- *Tình huống*: Công ty điện lực muốn cải thiện dịch vụ khách hàng.
- *Lợi ích*: Khách hàng có thể nhận hóa đơn ngay sau khi phát hành, kèm theo các tiện ích như nhắc nhở thanh toán, lịch sử hóa đơn, và các thông tin liên quan. Điều này nâng cao trải nghiệm và sự hài lòng của khách hàng.

2.4. Đối phó với tình huống khẩn cấp

- *Tình huống*: Đại dịch hoặc tình huống khẩn cấp khiến việc gửi hóa đơn giấy gặp khó khăn.
- *Lợi ích*: Hóa đơn điện tử không phụ thuộc vào việc gửi qua bưu điện, nên không bị ảnh hưởng bởi các yếu tố bên ngoài như thiên tai, dịch bệnh. Khách hàng vẫn có thể nhận hóa đơn và thanh toán bình thường.

3. Chức năng nghiệp vụ

- Quản lý thông tin khách hàng:
 - Thêm, sửa, và xóa thông tin của khách hàng trong bảng khách hàng.
- Quản lý thông tin công tơ điện:
 - Thêm, sửa, và xóa thông tin của các công tơ điện trong bảng công tơ điện.
- Ghi đo đếm tiêu thụ:
 - Ghi nhận mức tiêu thụ điện và lưu trữ vào bảng đo đếm tiêu thụ.

- Quản lý giá điện:

- Thêm, sửa, và xóa thông tin về các loại giá điện trong bảng giá điện (bao gồm: thời gian áp dụng, giá tiền cho 1 kWh).

- Tính toán tiền điện:

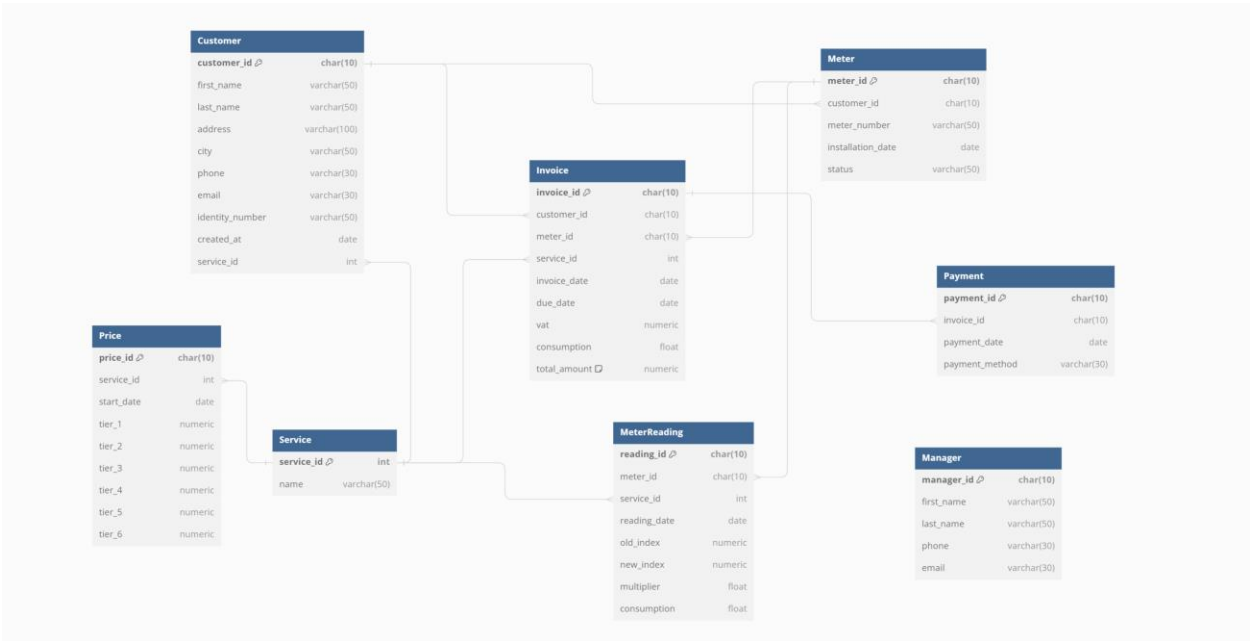
- Tính toán tiền điện phải trả dựa trên mức độ tiêu thụ điện và giá điện áp dụng.
- Tạo hóa đơn và lưu thông tin vào bảng hóa đơn.

- Quản lý hóa đơn:

- Xem, tìm kiếm, và xóa thông tin về hóa đơn trong bảng hóa đơn.
- Tạo hóa đơn mới cho khách hàng.

II. Cấu trúc dữ liệu

1. Sơ đồ thực thể liên kết:



- Mối quan hệ giữa các bảng:

1. One-to-Many (Một-đến-Nhiều)

- **Customer - Meter:** Một khách hàng có thể có nhiều công tơ (qua khóa ngoại customer_id trong bảng Meter tham chiếu đến khóa chính customer_id trong bảng Customer).
- **Service - Price:** Một dịch vụ có thể có nhiều mức giá khác nhau theo thời gian (qua khóa ngoại service_id trong bảng Price tham chiếu đến khóa chính service_id trong bảng Service).
- **Customer – Invoice:** Một khách hàng có thể có nhiều hóa đơn (qua khóa ngoại customer_id trong bảng Invoice tham chiếu đến khóa chính customer_id trong bảng Customer).
- **Meter - MeterReading:** Một công tơ có thể có nhiều chỉ số ghi nhận theo thời gian (qua khóa ngoại meter_id trong bảng MeterReading tham chiếu đến khóa chính meter_id trong bảng Meter).
- **Service - MeterReading:** Một dịch vụ được sử dụng bởi nhiều công tơ khác nhau (qua khóa ngoại service_id trong bảng MeterReading tham chiếu đến khóa chính service_id trong bảng Service).
- **Invoice – Payment:** Một hóa đơn có thể được thanh toán nhiều lần (qua khóa ngoại invoice_id trong bảng Payment tham chiếu đến khóa chính invoice_id trong bảng Invoice).

2. Many-to-Many (Nhiều-đến-Nhiều) (Gián tiếp):

- **Customer – Service:** Mặc dù không có trực tiếp khóa ngoại giữa hai bảng này, nhưng mối quan hệ nhiều-đến-nhiều được thiết lập gián tiếp thông qua các bảng Meter và Invoice. Bằng cách tham chiếu customer_id đến bảng Meter và meter_id đến bảng Invoice, cùng với service_id trong cả bảng MeterReading và Invoice, chúng ta có thể xác định dịch vụ nào được sử dụng bởi một khách hàng cụ thể.

3. Không có mối quan hệ (Independent - Độc lập)

- **Manager:** Bảng Manager không có mối quan hệ trực tiếp với các bảng khác trong ngữ cảnh được cung cấp.

2. Dữ liệu, yêu cầu, mối quan hệ

- Cấu trúc dữ liệu:

Bảng Dịch vụ (Service):

```
CREATE TABLE Service (  
  service_id INT NOT NULL,  
  name VARCHAR(50) NOT NULL,  
  CONSTRAINT Service_pk PRIMARY KEY (service_id)  
);
```

service_id	INT	Mã dịch vụ (PK)
name	VARCHAR(50)	Tên dịch vụ

Bảng Khách hàng (Customer):

```
CREATE TABLE Customer (  
  customer_id CHAR(10) NOT NULL,  
  first_name VARCHAR(50) NOT NULL,  
  last_name VARCHAR(50) NOT NULL,  
  address VARCHAR(100) NOT NULL,  
  city VARCHAR(50) NOT NULL,  
  phone VARCHAR(30) NOT NULL,  
  email VARCHAR(30) NOT NULL,  
  identity_number VARCHAR(50) NOT NULL UNIQUE,  
  created_at DATE,  
  service_id INT NOT NULL,  
  CONSTRAINT Customer_pk PRIMARY KEY (customer_id),  
  CONSTRAINT Customer_fk_Service FOREIGN KEY (service_id) REFERENCES Service(service_id)  
);
```

customer_id	CHAR(10)	Mã khách hàng (PK)
first_name	VARCHAR(50)	Họ
last_name	VARCHAR(50)	Tên
address	VARCHAR(100)	Địa chỉ
city	VARCHAR(50)	Thành phố
phone	VARCHAR(30)	Số điện thoại
email	VARCHAR(30)	Email
identity_number	VARCHAR(50)	Số nhận dạng cá nhân
created_at	DATE	Ngày tạo
service_id	INT	Mã dịch vụ (FK)

Bảng Quản lí (Manager):

```
CREATE TABLE Manager (  
  manager_id CHAR(10) NOT NULL,  
  first_name VARCHAR(50) NOT NULL,  
  last_name VARCHAR(50) NOT NULL,  
  phone VARCHAR(30) NOT NULL,  
  email VARCHAR(30) NOT NULL,  
  CONSTRAINT Manager_pk PRIMARY KEY (manager_id)  
);
```

manager_id	CHAR(10)	Mã quản lí (PK)
first_name	VARCHAR(50)	Họ
last_name	VARCHAR(50)	Tên
phone	VARCHAR(30)	Số điện thoại
email	VARCHAR(30)	Email

Bảng Điện kế (Meter):

```
CREATE TABLE Meter (  
  meter_id CHAR(10) NOT NULL,  
  customer_id CHAR(10) NOT NULL,  
  meter_number VARCHAR(50) NOT NULL,  
  installation_date DATE NOT NULL,  
  status VARCHAR(50) NOT NULL,  
  CONSTRAINT Meter_pk PRIMARY KEY (meter_id),  
  CONSTRAINT Meter_fk_Customer FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)  
);
```

meter_id	CHAR(10)	Mã điện kế (PK)
customer_id	CHAR(10)	Mã khách hàng (FK)
meter_number	VARCHAR(50)	Số hiệu điện kế
installation_date	DATE	Ngày lắp đặt
status	VARCHAR(50)	Trạng thái

Bảng Chỉ số điện (MeterReading):

```
CREATE TABLE MeterReading (  
  reading_id CHAR(10) NOT NULL,  
  meter_id CHAR(10) NOT NULL,  
  service_id INT NOT NULL,  
  reading_date DATE NOT NULL,  
  old_index NUMERIC,  
  new_index NUMERIC,  
  multiplier FLOAT,  
  consumption FLOAT,  
  CONSTRAINT MeterReading_pk PRIMARY KEY (reading_id),  
  CONSTRAINT MeterReading_fk_Meter FOREIGN KEY (meter_id) REFERENCES Meter(meter_id),  
  CONSTRAINT MeterReading_fk_Service FOREIGN KEY (service_id) REFERENCES Service(service_id)  
);
```

reading_id	CHAR(10)	Mã chỉ số điện (PK)
meter_id	CHAR(10)	Mã điện kế (FK)
service_id	INT	Mã dịch vụ (FK)
reading_date	DATE	Ngày ghi chỉ số
old_index	NUMERIC	Chỉ số cũ
new_index	NUMERIC	Chỉ số mới
multiplier	FLOAT	Cấp số nhân
consumption	FLOAT	Lượng điện tiêu thụ

Bảng Giá điện (Price):

```
CREATE TABLE Price (  
  price_id CHAR(10) NOT NULL,  
  service_id INT NOT NULL,  
  start_date DATE,  
  tier_1 NUMERIC,  
  tier_2 NUMERIC,  
  tier_3 NUMERIC,  
  tier_4 NUMERIC,  
  tier_5 NUMERIC,  
  tier_6 NUMERIC,  
  CONSTRAINT Price_pk PRIMARY KEY (price_id),  
  CONSTRAINT Price_fk_Service FOREIGN KEY (service_id) REFERENCES Service(service_id)  
);
```

price_id	CHAR(10)	Mã giá điện (PK)
service_id	INT	Mã dịch vụ (FK)
start_date	DATE	Ngày bắt đầu hiệu lực
tier_1	NUMERIC	Giá bậc 1
tier_2	NUMERIC	Giá bậc 2
tier_3	NUMERIC	Giá bậc 3
tier_4	NUMERIC	Giá bậc 4
tier_5	NUMERIC	Giá bậc 5
tier_6	NUMERIC	Giá bậc 6

Bảng Hóa đơn (Invoice):

```
CREATE TABLE Invoice (  
    invoice_id CHAR(10) NOT NULL,  
    customer_id CHAR(10) NOT NULL,  
    meter_id CHAR(10) NOT NULL,  
    service_id INT NOT NULL,  
    invoice_date DATE,  
    due_date DATE,  
    vat NUMERIC,  
    consumption FLOAT,  
    CONSTRAINT Invoice_pk PRIMARY KEY (invoice_id),  
    CONSTRAINT Invoice_fk_Customer FOREIGN KEY (customer_id) REFERENCES Customer(customer_id),  
    CONSTRAINT Invoice_fk_Meter FOREIGN KEY (meter_id) REFERENCES Meter(meter_id),  
    CONSTRAINT Invoice_fk_Service FOREIGN KEY (service_id) REFERENCES Service(service_id)  
);
```

invoice_id	CHAR(10)	Mã hóa đơn (PK)
customer_id	CHAR(10)	Mã khách hàng (FK)
meter_id	CHAR(10)	Mã điện kế (FK)
service_id	CHAR(10)	Mã dịch vụ (FK)
invoice_date	DATE	Ngày tạo hóa đơn
due_date	DATE	Ngày hết hạn hóa đơn
vat	NUMERIC	Thuế vat
consumption	FLOAT	Lượng điện tiêu thụ

Bảng Thanh toán (Payment):

```
CREATE TABLE Payment (  
    payment_id CHAR(10) NOT NULL,  
    invoice_id CHAR(10) NOT NULL,  
    payment_date DATE,  
    payment_method VARCHAR(30),  
    status VARCHAR(50) NOT NULL,  
    CONSTRAINT Payments_pk PRIMARY KEY (payment_id),  
    CONSTRAINT Payments_fk_Invoice FOREIGN KEY (invoice_id) REFERENCES Invoice(invoice_id)  
);
```

payment_id	CHAR(10)	Mã thanh toán (PK)
invoice_id	CHAR(10)	Mã hóa đơn (FK)
payment_date	DATE	Ngày thanh toán
payment_method	VARCHAR(30)	Phương thức thanh toán
status	VARCHAR(50)	Tình trạng thanh toán

3. Truy vấn SQL

3.1. Function và Trigger

- Hàm gửi email thông báo cho khách hàng khi có hóa đơn mới:

```
CREATE OR REPLACE FUNCTION notify_customer_on_new_invoice()
RETURNS TRIGGER AS $$
DECLARE
    v_email VARCHAR(30);
BEGIN
    -- Lấy địa chỉ email của khách hàng từ bảng Customer
    SELECT email INTO v_email FROM Customer WHERE customer_id = NEW.customer_id;

    -- Gửi email thông báo về hóa đơn mới được tạo
    PERFORM send_email(v_email, 'New Invoice Created', 'Your new invoice has been created with ID: ' || NEW.invoice_id || '.');

    -- Trả về bản ghi mới được chèn
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

-- Tạo trigger để kích hoạt hàm notify_customer_on_new_invoice sau khi chèn một bản ghi mới vào bảng Invoice
CREATE TRIGGER trg_notify_customer_on_new_invoice
AFTER INSERT ON Invoice
FOR EACH ROW
EXECUTE FUNCTION notify_customer_on_new_invoice();
```

Cách hoạt động:

- Hàm lấy địa chỉ email của khách hàng từ bảng ‘Customer’ dựa trên ‘customer_id’ của hóa đơn mới được tạo, rồi sử dụng hàm ‘send_email’ để gửi email thông báo cho khách hàng.
- Trigger sẽ kích hoạt hàm sau khi một bản ghi mới được chèn vào bảng ‘Invoice’

- Lấy hóa đơn chưa thanh toán của một khách hàng cụ thể.

```
CREATE OR REPLACE FUNCTION get_unpaid_invoice(p_customer_id CHAR(10))
RETURNS TABLE(invoice_id CHAR(10), invoice_date DATE, due_date DATE, total_amount NUMERIC) AS $$
BEGIN
    RETURN QUERY
    SELECT i.invoice_id, i.invoice_date, i.due_date, i.total_amount
    FROM Invoice i
    WHERE i.customer_id = p_customer_id
    AND i.invoice_id NOT IN (SELECT p.invoice_id FROM Payment p WHERE p.status = 'Đã thanh toán');
END;
$$ LANGUAGE plpgsql;
```

Cách hoạt động:

- Tìm tất cả các hóa đơn có ‘customer_id’ trùng với ‘p_customer_id’
- Lọc ra những hóa đơn có ‘invoice_id’ không nằm trong danh sách ‘Đã thanh toán’
- Trả về kết quả là bảng chứa các cột ‘invoice_id’, ‘invoice_date’, ‘due_date’ và ‘total_amount’.

- Tạo cột 'total_amount' cho bảng 'Invoice' để tính toán và tự động cập nhật tổng tiền phải trả của mỗi bản ghi hóa đơn.

```
ALTER TABLE Invoice ADD COLUMN total_amount NUMERIC DEFAULT 0;

CREATE OR REPLACE FUNCTION calculate_total_amount()
RETURNS TRIGGER AS $$
DECLARE
    v_tier_1 NUMERIC;
    v_tier_2 NUMERIC;
    v_tier_3 NUMERIC;
    v_tier_4 NUMERIC;
    v_tier_5 NUMERIC;
    v_tier_6 NUMERIC;
    v_consumption FLOAT;
    v_vat NUMERIC;
BEGIN
    SELECT tier_1, tier_2, tier_3, tier_4, tier_5, tier_6
    INTO v_tier_1, v_tier_2, v_tier_3, v_tier_4, v_tier_5, v_tier_6
    FROM Price
    WHERE service_id = NEW.service_id;

    v_consumption := NEW.consumption;
    v_vat := NEW.vat;

    IF v_consumption <= 50 THEN
        NEW.total_amount := v_consumption * v_tier_1 * (1 + v_vat);
    ELSIF v_consumption <= 100 THEN
        NEW.total_amount := (50 * v_tier_1 + (v_consumption - 50) * v_tier_2) * (1 + v_vat);
    ELSIF v_consumption <= 200 THEN
        NEW.total_amount := (50 * v_tier_1 + 50 * v_tier_2 + (v_consumption - 100) * v_tier_3) * (1 + v_vat);
    ELSIF v_consumption <= 300 THEN
        NEW.total_amount := (50 * v_tier_1 + 50 * v_tier_2 + 100 * v_tier_3 + (v_consumption - 200) * v_tier_4) * (1 + v_vat);
    ELSIF v_consumption <= 400 THEN
        NEW.total_amount := (50 * v_tier_1 + 50 * v_tier_2 + 100 * v_tier_3 + 100 * v_tier_4 + (v_consumption - 300) * v_tier_5) * (1 + v_vat);
    ELSE
        NEW.total_amount := (50 * v_tier_1 + 50 * v_tier_2 + 100 * v_tier_3 + 100 * v_tier_4 + 100 * v_tier_5 + (v_consumption - 400) * v_tier_6) * (1 + v_vat);
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER trg_calculate_total_amount
BEFORE INSERT ON Invoice
FOR EACH ROW
EXECUTE FUNCTION calculate_total_amount();
```

Cách hoạt động:

- Tạo thêm cột 'total_amount' cho bảng 'Invoice'.
- Hàm function sẽ tính giá tiền của một hóa đơn theo công thức sau:

TT	Nhóm đối tượng khách hàng	Giá bán điện (đồng/kWh)
1	Giá bán lẻ điện cho sinh hoạt	
1.1	Giá bán lẻ điện sinh hoạt	
	Bậc 1: Cho kWh từ 0 - 50	1.806
	Bậc 2: Cho kWh từ 51 - 100	1.866
	Bậc 3: Cho kWh từ 101 - 200	2.167
	Bậc 4: Cho kWh từ 201 - 300	2.729
	Bậc 5: Cho kWh từ 301 - 400	3.050
	Bậc 6: Cho kWh từ 401 trở lên	3.151
1.2	Giá bán lẻ điện sinh hoạt dùng công tơ thẻ trả trước	2.649

- Trigger sẽ kích hoạt hàm trước khi thông tin của hóa đơn được đưa vào bảng 'Invoice'.

3.2. View:

- View để xem thông tin chi tiết khách hàng

```
-- View để xem thông tin chi tiết khách hàng
CREATE OR REPLACE VIEW vw_customer_details AS
SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    c.address,
    c.city,
    c.phone,
    c.email,
    c.identity_number,
    c.created_at,
    s.name AS service_name
FROM
    Customer c
JOIN
    Service s ON c.service_id = s.service_id;
```

- View để xem chỉ số công tơ mới nhất cho mỗi khách hàng.

```
CREATE OR REPLACE VIEW vw_latest_meter_reading AS
SELECT
    m.customer_id,
    m.meter_id,
    m.meter_number,
    mr.reading_date,
    mr.new_index
FROM
    Meter m
JOIN
    MeterReading mr ON m.meter_id = mr.meter_id
WHERE
    mr.reading_date = (
        SELECT MAX(reading_date)
        FROM MeterReading mr_sub
        WHERE mr_sub.meter_id = m.meter_id
    );
```

- View để xem các hóa đơn chưa thanh toán của khách hàng.

```
CREATE OR REPLACE VIEW vw_unpaid_invoices AS
SELECT
    i.invoice_id,
    i.customer_id,
    c.first_name,
    c.last_name,
    i.invoice_date,
    i.due_date,
    i.total_amount
FROM
    Invoice i
JOIN
    Customer c ON i.customer_id = c.customer_id
LEFT JOIN
    Payment p ON i.invoice_id = p.invoice_id
WHERE
    p.invoice_id IS NULL OR p.status <> 'Đã thanh toán';
```

3.3. Query:

1. Lấy thông tin chi tiết về một quản lí cụ thể:

```
SELECT * FROM Manager WHERE manager_id = 'M1';
```

	manager_id [PK] character	first_name character varying (50)	last_name character varying (50)	phone character varying (30)	email character varying (30)
1	M1	Trần	Phương Thúy	0123453789	tranphngthuy@example.com

2. Tính tổng mức tiêu thụ của khách hàng cụ thể:

```
SELECT customer_id, SUM(consumption) AS total_consumption  
FROM Invoice  
WHERE customer_id = 'C1'  
GROUP BY customer_id;
```

	customer_id character	total_consumption double precision
1	C1	650

3. Lấy thông tin chi tiết về các khách hàng sử dụng dịch vụ cụ thể:

```
SELECT c.customer_id, c.first_name, c.last_name, c.email, c.phone  
FROM Customer c  
WHERE c.service_id = 1;
```

	customer_id [PK] character	first_name character varying (50)	last_name character varying (50)	email character varying (30)	phone character varying (30)
1	C1	Nguyễn	Văn Anh	nguyenvananh@example.com	0987654321
2	C3	Lê	Văn Cường	levancuong@example.com	0978123456
3	C5	Nguyễn	Văn Đức	nguyenvanduc@example.com	0965432109
4	C7	Võ	Văn Giang	vovang@example.com	0909876543
5	C9	Lê	Văn Hùng	levanhung@example.com	0970123456

4. Hiển thị các khách hàng có địa chỉ trong một thành phố cụ thể:

```
SELECT * FROM Customer  
WHERE city = 'Hà Nội';
```

	customer_id [PK] character	first_name character varying (50)	last_name character varying (50)	address character varying (100)	city character varying (50)	phone character varying (30)
1	C1	Nguyễn	Văn Anh	123 Đường Hoàng Văn Thụ	Hà Nội	0987654321
2	C7	Võ	Văn Giang	123 Đường Nguyễn Văn Linh	Hà Nội	0909876543
3	C12	Mai	Thị Mến	654 Đường Ngô Quyền	Hà Nội	0987654321
4	C29	Mai	Thị Dung	123 Đường Hoàng Văn Thụ	Hà Nội	0987654321

5. Tính tổng tiền thanh toán cho mỗi phương thức thanh toán:

```
SELECT payment_method, SUM(Invoice.total_amount) AS total_paid
FROM Payment
INNER JOIN Invoice ON Payment.invoice_id = Invoice.invoice_id
GROUP BY payment_method;
```

	payment_method character varying (30)	total_paid numeric
1	Trực tiếp	45594.494
2	Trực tuyến	59342.0168

6. Lấy tất cả các dịch vụ và giá cả tương ứng của chúng:

```
SELECT * FROM Service
INNER JOIN Price ON Service.service_id = Price.service_id;
```

	service_id integer	name character varying (50)	price_id character	service_id integer	start_date date	tier_1 numeric	tier_2 numeric	tier_3 numeric	tier_4 numeric	tier_5 numeric
1	1	Điện sinh hoạt	P01	1	2023-11-09	1.806	1.866	2.167	2.729	3.050
2	2	Điện kinh doanh	P02	2	2023-11-09	1.999	2.103	2.312	2.897	3.192

7. Liệt kê tất cả các hóa đơn với mức tiêu thụ vượt quá một ngưỡng nhất định:

```
SELECT
    i.invoice_id,
    i.customer_id,
    i.invoice_date,
    i.total_amount,
    i.consumption
FROM
    Invoice i
WHERE
    i.consumption > 700
ORDER BY
    i.consumption DESC;
```

	invoice_id [PK] character	customer_id character	invoice_date date	total_amount numeric	consumption double precision
1	I23	C23	2024-01-23	3502.29	1100
2	I73	C73	2024-06-29	3502.29	1100
3	I77	C77	2024-07-02	2372.3414	774
4	I27	C27	2024-01-27	2372.3414	774

8. Liệt kê tất cả các bộ đo đã lắp đặt trong khoảng thời gian cụ thể:

```
SELECT * FROM Meter
WHERE installation_date BETWEEN '2022-01-01' AND '2022-05-15';
```

	meter_id [PK] character	customer_id character	meter_number character varying (50)	installation_date date	status character varying (50)
1	MT25	C25	Z0E1Y3W8	2022-01-06	Active
2	MT26	C26	K4N6U9Y2	2022-02-15	Active
3	MT27	C27	P3S6W9X5	2022-03-31	Active
4	MT28	C28	T9V1A3C2	2022-04-26	Active
5	MT29	C29	L6Q8F3E7	2022-05-11	Active

9. Liệt kê tất cả các khách hàng và số lượng hóa đơn mà họ đã tạo ra:

```
SELECT Customer.customer_id,
COUNT(Invoice.invoice_id) AS invoice_count
FROM Customer
LEFT JOIN Invoice ON Customer.customer_id = Invoice.customer_id
GROUP BY Customer.customer_id
ORDER BY Customer.customer_id;
```

	customer_id [PK] character	invoice_count bigint
1	C1	1
2	C10	1
3	C100	1
4	C11	1

10. Tìm tất cả các hóa đơn mà tỷ lệ tiêu thụ vượt quá mức nhất định:

```
SELECT * FROM Invoice
WHERE consumption > (SELECT AVG(consumption) FROM Invoice);
```

	invoice_id [PK] character	customer_id character	meter_id character	service_id integer	invoice_date date	due_date date	vat numeric	consumption double precision	total_amount numeric
1	I1	C1	MT1	1	2024-05-14	2024-06-14	0.1	650	1942.545
2	I3	C3	MT3	1	2024-05-14	2024-06-14	0.1	500	1422.63
3	I6	C6	MT6	2	2024-05-16	2024-06-16	0.1	750	2448.71
4	I10	C10	MT10	2	2024-05-24	2024-06-24	0.1	530	1632.202
5	I11	C11	MT11	1	2024-05-26	2024-06-26	0.1	750	2280.155

11. Liệt kê tất cả các hóa đơn với mức tiêu thụ tăng dần:

SELECT * FROM Invoice ORDER BY consumption ASC;

	invoice_id [PK] character	customer_id character	meter_id character	service_id integer	invoice_date date	due_date date	vat numeric	consumption double precision	total_amount numeric
1	I21	C21	MT21	1	2024-06-01	2024-07-01	0.1	100	201.96
2	I7	C7	MT7	1	2024-05-15	2024-06-18	0.1	100	201.96
3	I8	C8	MT8	2	2024-05-20	2024-06-20	0.1	100	225.61
4	I29	C29	MT29	1	2024-06-06	2024-07-06	0.1	100	201.96
5	I26	C26	MT26	2	2024-06-06	2024-07-06	0.1	100	225.61
6	I22	C22	MT22	2	2024-06-02	2024-07-02	0.1	100	225.61
7	I19	C19	MT19	1	2024-06-06	2024-07-06	0.1	100	201.96
8	I2	C2	MT2	2	2024-05-14	2024-06-14	0.1	120	276.474
9	I32	C32	MT2	2	2024-06-09	2024-07-09	0.1	150	352.77
10	I46	C46	MT16	2	2024-06-23	2024-07-23	0.1	150	352.77
11	I5	C5	MT5	1	2024-05-14	2024-06-14	0.1	150	321.145
12	I38	C38	MT8	2	2024-06-15	2024-07-15	0.1	150	352.77

12. Liệt kê tất cả các khách hàng đã sử dụng dịch vụ trong khoảng thời gian cụ thể:

SELECT DISTINCT Customer.customer_id, Customer.first_name, Customer.last_name
FROM Customer
INNER JOIN Invoice ON Customer.customer_id = Invoice.customer_id
WHERE Invoice.invoice_date BETWEEN '2023-01-01' AND '2024-12-31';

	customer_id [PK] character	first_name character varying (50)	last_name character varying (50)
1	C40	Nguyễn	Văn Phúc
2	C64	Lê	Văn Hiếu
3	C89	Trần	Thị Thanh
4	C47	Võ	Thị Xuyên
5	C12	Mai	Thị Mến

13. Hiển thị tất cả các bộ đo và số đếm cho mỗi loại dịch vụ:

SELECT Service.name AS service_name, Meter.meter_id, Meter.meter_number
FROM Meter
INNER JOIN Customer ON Meter.customer_id = Customer.customer_id
INNER JOIN Service ON Customer.service_id = Service.service_id;

	service_name character varying (50)	meter_id character	meter_number character varying (50)
1	Điện sinh hoạt	MT1	MT45K9Q8
2	Điện kinh doanh	MT2	M5Y8D0A2
3	Điện sinh hoạt	MT3	MZ4K6A0E
4	Điện kinh doanh	MT4	8Q7M3B6H

14. Xem tất cả các hóa đơn cho một khách hàng cụ thể:

SELECT * FROM Invoice WHERE customer_id = 'C1';

	invoice_id [PK] character	customer_id character	meter_id character	service_id integer	invoice_date date	due_date date	vat numeric	consumption double precision	total_amount numeric
1	I1	C1	MT1	1	2024-05-14	2024-06-14	0.1	650	1942.545

15. Liệt kê tất cả các hóa đơn với tiền thanh toán vượt qua một ngưỡng nhất định:

SELECT * FROM Invoice WHERE total_amount > '1200';

	invoice_id [PK] character	customer_id character	meter_id character	service_id integer	invoice_date date	due_date date	vat numeric	consumption double precision	total_amount numeric
1	I1	C1	MT1	1	2024-05-14	2024-06-14	0.1	650	1942.545
2	I3	C3	MT3	1	2024-05-14	2024-06-14	0.1	500	1422.63
3	I6	C6	MT6	2	2024-05-16	2024-06-16	0.1	750	2448.71
4	I10	C10	MT10	2	2024-05-24	2024-06-24	0.1	530	1632.202
5	I11	C11	MT11	1	2024-05-26	2024-06-26	0.1	750	2289.155
6	I13	C13	MT13	1	2024-05-30	2024-06-30	0.1	690	2081.189

16. Tính tổng lượng tiêu thụ của mỗi loại dịch vụ:

SELECT Service.name,
SUM(Invoice.consumption)
AS total_consumption
FROM Invoice
INNER JOIN Service
ON Invoice.service_id = Service.service_id
GROUP BY Service.name;

	name character varying (50)	total_consumption double precision
1	Điện sinh hoạt	21098
2	Điện kinh doanh	16020

17. Hiển thị tất cả các hóa đơn phát sinh từ một ngày cụ thể đến ngày hiện tại:

SELECT * FROM Invoice
WHERE invoice_date
BETWEEN '2024-05-15' AND CURRENT_DATE;

	invoice_id [PK] character	customer_id character	meter_id character	service_id integer	invoice_date date	due_date date	vat numeric	consumption double precision	total_amount numeric
1	I55	C55	MT55	1	2024-05-26	2024-06-26	0.1	200	440.33
2	I60	C60	MT60	2	2024-06-16	2024-07-16	0.1	190	454.498
3	I84	C84	MT84	2	2024-06-10	2024-07-10	0.1	350	974.16
4	I97	C97	MT97	1	2024-06-07	2024-07-07	0.1	400	1076.02

18. Liệt kê tất cả các khách hàng và số lượng hóa đơn đã thanh toán:

```
SELECT Customer.customer_id, Customer.first_name, Customer.last_name,
COUNT(Payment.invoice_id) AS paid_invoice_count
FROM Customer
INNER JOIN Invoice ON Customer.customer_id = Invoice.customer_id
INNER JOIN Payment ON Invoice.invoice_id = Payment.invoice_id
GROUP BY Customer.customer_id, Customer.first_name, Customer.last_name;
```

	customer_id [PK] character	first_name character varying (50)	last_name character varying (50)	paid_invoice_count bigint
1	C70	Phạm	Văn Phát	1
2	C67	Nguyễn	Thị Lan	1
3	C85	Trần	Thị Ngọc	1
4	C53	Trần	Thị Diệu	1

19. Liệt kê tất cả các khách hàng và thông tin liên lạc của họ:

```
SELECT customer_id, first_name, last_name, address, city, phone, email FROM Customer;
```

	customer_id [PK] character	first_name character varying (50)	last_name character varying (50)	address character varying (100)	city character varying (50)	phone character varying (30)
1	C1	Nguyễn	Văn Anh	123 Đường Hoàng Văn Thụ	Hà Nội	0987654321
2	C2	Trần	Thị Bảo	456 Đường Lê Lợi	Hồ Chí Minh	0901234567
3	C3	Lê	Văn Cường	789 Đường Nguyễn Huệ	Đà Nẵng	0978123456
4	C4	Phạm	Thị Diệu	321 Đường Trần Hưng Đạo	Cần Thơ	0912345678
5	C5	Nguyễn	Văn Đức	654 Đường Lê Lai	Hải Phòng	0965432109

20. Xem tất cả các đọc số điện mới nhất và cũ nhất:

```
SELECT
    c.customer_id, c.first_name, c.last_name, r.old_index, r.new_index, r.reading_date
FROM
    Customer c
JOIN
    Meter m ON c.customer_id = m.customer_id
JOIN
    MeterReading r ON m.meter_id = r.meter_id
ORDER BY
    c.customer_id, r.reading_date;
```

	customer_id character	first_name character varying (50)	last_name character varying (50)	old_index numeric	new_index numeric	reading_date date
1	C1	Nguyễn	Văn Anh	1200	1850	2024-01-01
2	C10	Phan	Thị Khánh	1230	1760	2024-01-09
3	C100	Lê	Văn Minh	900	1150	2024-06-25
4	C11	Hoàng	Văn Long	2700	3450	2024-01-10
5	C12	Mai	Thị Mến	2200	2600	2024-01-11

21. Tìm tất cả các hóa đơn phát sinh sau một ngày nhất định:

SELECT * FROM Invoice WHERE invoice_date > '2024-06-25';

	invoice_id [PK] character	customer_id character	meter_id character	service_id integer	invoice_date date	due_date date	vat numeric	consumption double precision	total_amount numeric
1	I52	C52	MT52	2	2024-07-20	2024-08-20	0.1	130	301.906
2	I53	C53	MT53	1	2024-09-29	2024-10-29	0.1	450	1249.325
3	I56	C56	MT56	2	2024-11-11	2024-12-11	0.1	750	2448.71
4	I58	C58	MT58	2	2024-08-05	2024-09-05	0.1	100	225.61
5	I61	C61	MT61	1	2024-10-31	2024-11-30	0.1	650	1942.545

22. Lấy danh sách các hóa đơn đã hết hạn:

```

SELECT
    i.invoice_id,
    c.customer_id,
    c.first_name,
    c.last_name,
    i.invoice_date,
    i.due_date,
    i.total_amount
FROM
    Invoice i
JOIN
    Customer c ON i.customer_id = c.customer_id
LEFT JOIN
    Payment p ON i.invoice_id = p.invoice_id
WHERE
    i.due_date < CURRENT_DATE AND (p.status != 'Đã thanh toán')
ORDER BY
    i.due_date DESC;
```

	invoice_id character	customer_id character	first_name character varying (50)	last_name character varying (50)	invoice_date date	due_date date	total_amount numeric
1	I59	C59	Lê	Thị Bình	2024-04-24	2024-05-24	368.819
2	I81	C81	Trần	Thị Hương	2024-04-16	2024-05-16	1942.545
3	I42	C42	Đỗ	Văn Rạng	2024-04-02	2024-05-02	479.93
4	I87	C87	Nguyễn	Thị Quý	2024-04-02	2024-05-02	908.27
5	I39	C39	Lê	Thị Oanh	2024-03-21	2024-04-21	590.425

23. Liệt kê mức tiêu thụ của các khách hàng trong một tháng cụ thể:

```
SELECT
  c.customer_id,
  c.first_name,
  c.last_name,
  SUM(i.consumption) AS total_consumption
FROM
  Customer c
JOIN
  Invoice i ON c.customer_id = i.customer_id
WHERE
  DATE_TRUNC('month', i.invoice_date) = '2024-05-01'
GROUP BY
  c.customer_id, c.first_name, c.last_name
ORDER BY
  total_consumption DESC
LIMIT 10;
```

	customer_id [PK] character	first_name character varying (50)	last_name character varying (50)	total_consumption double precision
1	C70	Phạm	Văn Phát	760
2	C55	Phạm	Thị Đào	200

24. Lấy thông tin các hóa đơn chưa thanh toán:

```
SELECT
  i.invoice_id,
  i.customer_id,
  i.invoice_date,
  i.due_date,
  i.total_amount
FROM
  Invoice i
LEFT JOIN
  Payment p ON i.invoice_id = p.invoice_id
WHERE
  p.status IS NULL OR p.status != 'Đã thanh toán';
```

	invoice_id [PK] character	customer_id character	invoice_date date	due_date date	total_amount numeric
1	I2	C2	2024-01-02	2024-02-02	276.474
2	I7	C7	2024-01-07	2024-02-07	201.96
3	I9	C9	2024-01-09	2024-02-09	841.17
4	I11	C11	2024-01-11	2024-02-11	2289.155
5	I13	C13	2024-01-13	2024-02-13	2081.189

25. Lấy danh sách các hóa đơn và ngày thanh toán tương ứng:

```
SELECT
    i.invoice_id,
    i.invoice_date,
    i.total_amount,
    p.payment_date
FROM
    Invoice i
LEFT JOIN
    Payment p ON i.invoice_id = p.invoice_id
ORDER BY
    i.invoice_date, p.payment_date;
```

	invoice_id character	invoice_date date	total_amount numeric	payment_date date
1	I1	2024-01-02	1942.545	2024-01-15
2	I2	2024-01-02	276.474	[null]
3	I3	2024-01-03	1422.63	2024-01-17
4	I4	2024-01-04	479.93	2024-02-18
5	I5	2024-01-05	321.145	2024-05-19

26. Đếm số lượng khách hàng sử dụng từng loại dịch vụ:

```
SELECT
    s.name AS service_name,
    COUNT(c.customer_id) AS customer_count
FROM
    Service s
JOIN
    Customer c ON s.service_id = c.service_id
GROUP BY
    s.name
ORDER BY
    customer_count DESC;
```

	service_name character varying (50)	customer_count bigint
1	Điện sinh hoạt	50
2	Điện kinh doanh	50

27. Lấy thông tin về các khách hàng đã thanh toán hóa đơn trước ngày đến hạn:

```
SELECT
    p.payment_id,
    p.payment_date,
    p.status,
    i.invoice_id,
    i.due_date,
    c.customer_id,
    c.first_name,
    c.last_name
FROM Payment p
JOIN Invoice i ON p.invoice_id = i.invoice_id
JOIN Customer c ON i.customer_id = c.customer_id
WHERE p.payment_date <= i.due_date
ORDER BY
    p.payment_date;
```

	payment_id character	payment_date date	status character varying (50)	invoice_id character	due_date date	customer_id character	first_name character varying (50)	last_name character varying (
1	P12	2024-01-11	Đã thanh toán	I12	2024-02-12	C12	Mai	Thị Mến
2	P1	2024-01-15	Đã thanh toán	I1	2024-02-02	C1	Nguyễn	Văn Anh
3	P3	2024-01-17	Đã thanh toán	I3	2024-02-03	C3	Lê	Văn Cường
4	P6	2024-01-20	Đã thanh toán	I6	2024-02-06	C6	Bùi	Thị Êm
5	P10	2024-01-24	Đã thanh toán	I10	2024-02-10	C10	Phan	Thị Khánh

28. Lấy danh sách các khách hàng kèm theo số lượng đồng hồ họ sử dụng:

```
SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    COUNT(m.meter_id) AS meter_count
FROM
    Customer c
JOIN
    Meter m ON c.customer_id = m.customer_id
GROUP BY
    c.customer_id, c.first_name, c.last_name
ORDER BY
    meter_count DESC;
```

	customer_id [PK] character	first_name character varying (50)	last_name character varying (50)	meter_count bigint
1	C70	Phạm	Văn Phát	1
2	C67	Nguyễn	Thị Lan	1
3	C85	Trần	Thị Ngọc	1
4	C53	Trần	Thị Diệu	1
5	C3	Lê	Văn Cường	1

29. Lấy thông tin về số lượng hóa đơn chưa thanh toán theo phương thức thanh toán:

```
SELECT
  COALESCE(p.payment_method, 'Không có') AS payment_method,
  COUNT(i.invoice_id) AS invoice_count,
  SUM(i.total_amount) AS total_amount
FROM
  Invoice i
LEFT JOIN
  Payment p ON i.invoice_id = p.invoice_id
WHERE
  p.status IS NULL OR p.status != 'Đã thanh toán'
GROUP BY
  payment_method
ORDER BY
  total_amount DESC;
```

	payment_method character varying	invoice_count bigint	total_amount numeric
1	Trực tuyến	46	54579.6768
2	Trực tiếp	5	5105.441

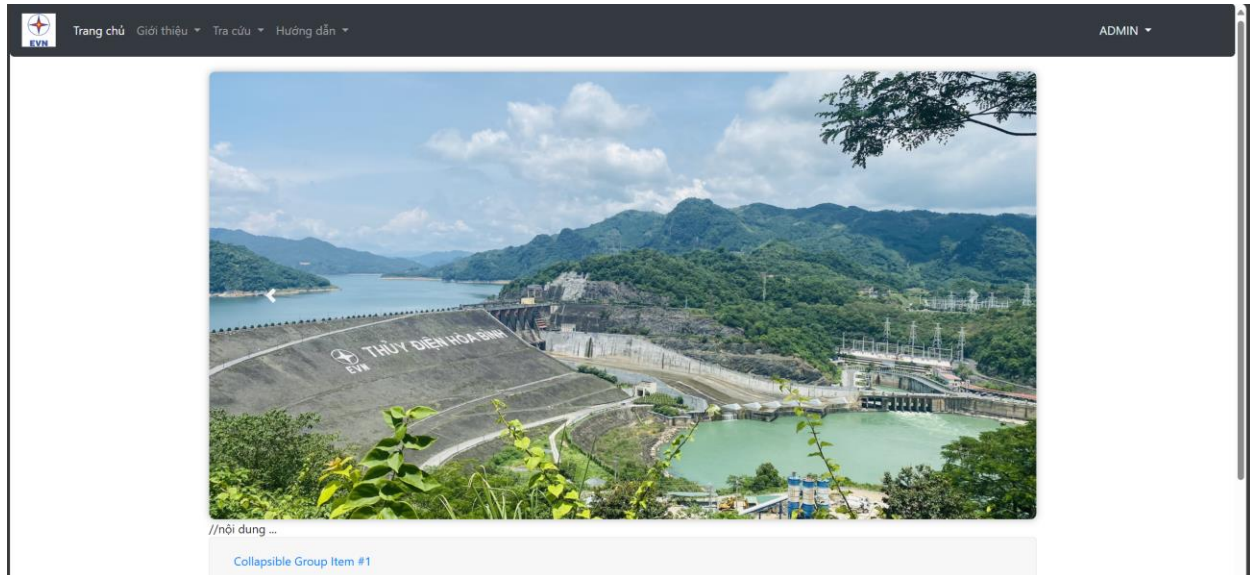
30. Lấy thông tin chi tiết của các quản lý và số lượng khách hàng mà họ quản lý:

```
ALTER TABLE Service ADD COLUMN manager_id CHAR(10);
UPDATE Service SET manager_id = 'M1' WHERE service_id = 1;
UPDATE Service SET manager_id = 'M2' WHERE service_id = 2;
SELECT
  m.manager_id,
  m.first_name,
  m.last_name,
  COUNT(c.customer_id) AS customer_count
FROM
  Manager m
JOIN
  Service s ON m.manager_id = s.manager_id
JOIN
  Customer c ON s.service_id = c.service_id
GROUP BY
  m.manager_id, m.first_name, m.last_name
ORDER BY
  customer_count DESC;
```

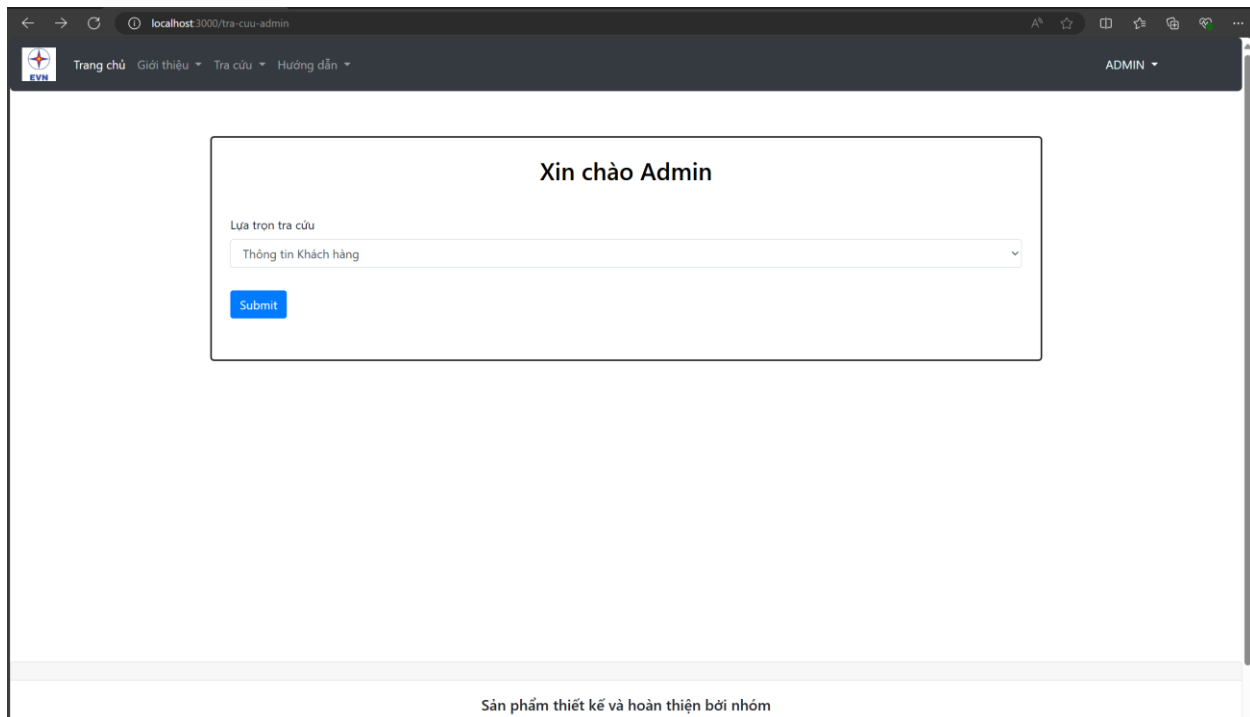
	manager_id [PK] character	first_name character varying (50)	last_name character varying (50)	customer_count bigint
1	M1	Trần	Phương Thủy	50
2	M2	Trịnh	Kim Bích	50

III. Kết quả:

Đây là giao diện của trang web sử dụng để quản lý hóa đơn tiền điện cho khách hàng.

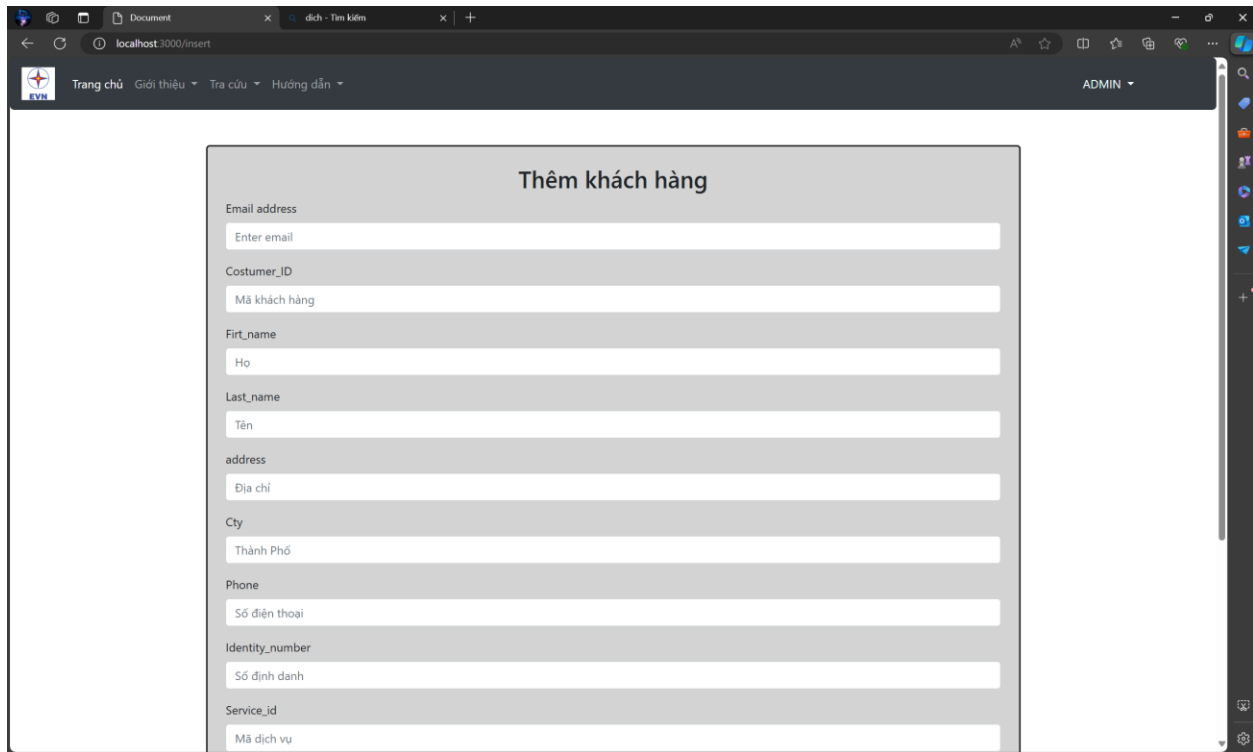


Trong giao diện này, người quản lý có thể dễ dàng thực hiện hiệu quả một số thao tác cơ bản. Dưới đây là giao diện khi thực hiện các thao tác trên web:



Một số thao tác cơ bản mà các quản lí có thể thực hiện lên trên cơ sở dữ liệu:

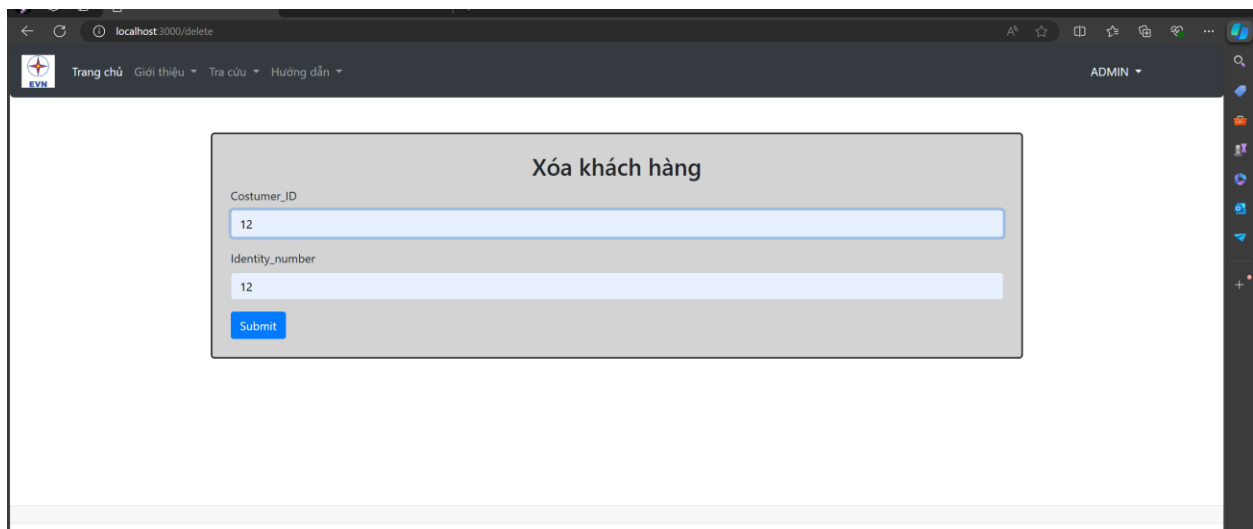
- Tạo thêm dữ liệu về các khách hàng mới:



The screenshot shows a web browser window with the URL `localhost:3000/insert`. The page has a dark header with a logo on the left and navigation links: `Trang chủ`, `Giới thiệu`, `Trà cứu`, and `Hướng dẫn`. On the right side of the header is the text `ADMIN` with a dropdown arrow. The main content area is a light gray box titled `Thêm khách hàng`. Inside this box is a form with the following fields: `Email address` (with placeholder `Enter email`), `Costumer_ID` (with placeholder `Mã khách hàng`), `Firt_name` (with placeholder `Họ`), `Last_name` (with placeholder `Tên`), `address` (with placeholder `Địa chỉ`), `Cty` (with placeholder `Thành Phố`), `Phone` (with placeholder `Số điện thoại`), `Identity_number` (with placeholder `Số định danh`), and `Service_id` (with placeholder `Mã dịch vụ`).

- Xóa dữ liệu của khách hàng.

Ví dụ, ở đây khi chúng ta muốn xóa dữ liệu của khách hàng có mã ID là 12 và số định danh (hay còn gọi là CCCD) là 12. Chúng ta chỉ việc nhập thông tin về 2 mục đó của khách hàng này, sau đó ấn submit. Trang web hiển thị ra dòng chữ “Customer deleted successfully”, khi đó chúng ta đã xóa dữ liệu về khách hàng có mã ID là 12 thành công.



The screenshot shows a web browser window with the URL `localhost:3000/delete`. The page has the same dark header as the previous screenshot, with the same navigation links and `ADMIN` text. The main content area is a light gray box titled `Xóa khách hàng`. Inside this box is a form with two fields: `Costumer_ID` (containing the value `12`) and `Identity_number` (containing the value `12`). Below these fields is a blue button labeled `Submit`.



- Kiểm tra danh sách các khách hàng đã đăng kí:

VĂN PHÒNG								
C41	Phạm	Thị Quỳnh	910 Đường Hoàng Văn Thụ	Hồ Chí Minh	0911223354	phamthiquynh@example.com	112567890123	1
C42	Đỗ	Văn Rang	112 Đường Lê Duẩn	Đà Nẵng	0911223355	dovanrang@example.com	113678901234	2
C43	Nguyễn	Thị Sang	345 Đường Nguyễn Huệ	Cần Thơ	0911223356	nguyenthisang@example.com	114789012345	1
C44	Trần	Văn Tâm	678 Đường Trần Hưng Đạo	Hà Nội	0911223357	tranvantam@example.com	115890123456	2
C45	Lê	Thị Uyên	910 Đường Ngô Quyền	Hải Phòng	0911223358	lethiuyen@example.com	116901234567	1
C46	Bùi	Văn Vinh	112 Đường Hồ Tùng Mậu	Nam Định	0911223359	buivanvinh@example.com	117012345678	2
C47	Võ	Thị Xuyên	345 Đường Lê Lợi	Hồ Chí Minh	0911223360	vothiuyen@example.com	118123456789	1
C48	Nguyễn	Văn Yên	678 Đường Nguyễn Văn Linh	Quảng Ninh	0911223361	nguyenvanyen@example.com	119234567890	2
C49	Phạm	Thị Ánh	910 Đường Trần Hưng Đạo	Đà Nẵng	0911223362	phamthianh@example.com	120345678901	1
C50	Lê	Văn Bình	112 Đường Lê Duẩn	Cần Thơ	0911223363	levanbinh@example.com	121456789012	2

- Tra cứu dữ liệu, thông tin về từng khách hàng cụ thể.

Ví dụ, các quản lí có thể truy vấn lịch sử thanh toán hóa đơn của người dùng và thông tin chi tiết về hóa đơn của người dùng đó dựa theo bảng Payment và Invoice.

A screenshot of a web browser window at localhost:3000/tracuu. The page has a dark header with a logo and navigation links. The main content area contains a search form with a text input for the customer ID (filled with "101234567890"), a dropdown menu for the search type (set to "lịch sử thanh toán"), and a blue "Submit" button.

Trang chủ Giới thiệu Tra cứu Hướng dẫn ADMIN

Nhập mã khách hàng

101234567890

Lựa chọn tra cứu

lịch sử thanh toán

Submit

EVN

Trang chủ

Giới thiệu

Tra cứu

Hướng dẫn

ADMIN

Kết quả tra cứu invoice của khách hàng

first_name	last_name	customer_id	identity_number	invoice_id	meter_id	service_id	invoice_date	due_date	vat	consumption	total amount
Nguyễn	Văn Anh	C1	101234567890	I1	MT1	1	2024-05-14	2024-06-14	0.1	650	1942.545
Nguyễn	Văn Anh	C1	101234567890	I31	MT1	1	2024-06-08	2024-07-08	0.1	650	1942.545

Thanh toán

payment_id	invoice_id	payment_date	payment_method
P1	I1	2024-05-15	Trực tuyến
P31	I31	2024-06-14	Trực tuyến

Tóm lại, giao diện của chúng tôi được thiết kế để các quản lí có thể thực hiện hiệu quả các thao tác cơ bản. Bên cạnh đó, các khách hàng cũng có thể thao tác dễ dàng để xem thông tin chi tiết về hóa đơn của bản thân trên trang web này. Hy vọng rằng hướng dẫn trên đã giúp bạn hiểu rõ hơn về cách sử dụng giao diện trang web của chúng tôi.

IV. Thành viên nhóm:

1. Vũ Quốc Bảo (TN):

- MSSV: 20225694
- Email: bao.vq225694@sis.hust.edu.vn
- Lớp: IT Việt Nhật 07, K67
- Nhiệm vụ: tao database, tao các hàm và câu truy vấn, viết báo cáo

2. Vương Quốc Huy:

- MSSV: 20225637
- Email: huy.vq225637@sis.hust.edu.vn
- Lớp: IT Việt Nhật 07, K67
- Nhiệm vụ: lên ý tưởng, tao câu truy vấn, làm powerpoint thuyết trình

3. Đỗ Trung Hiếu:

- MSSV: 20225623
- Email: hieu.dt225623@sis.hust.edu.vn
- Lớp: IT Việt Nhật 07, K67
- Nhiệm vụ: lên ý tưởng, tạo giao diện trang web