```sql
CREATE TABLE Users (
    user_id SERIAL PRIMARY KEY,
    username VARCHAR(50) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    role VARCHAR(20) CHECK (role IN ('superuser', 'admin', 'user')) NOT NULL
);

CREATE TABLE Products (
    product_id SERIAL PRIMARY KEY,
    product_name VARCHAR(100) NOT NULL,
    price NUMERIC(10, 2) NOT NULL,
    stock INT NOT NULL CHECK (stock >= 0)
);

CREATE TABLE Transactions (
    transaction_id SERIAL PRIMARY KEY,
    transaction_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    cashier_name VARCHAR(50) NOT NULL
);

CREATE TABLE Transaction_Details (
    detail_id SERIAL PRIMARY KEY,
    transaction_id INT REFERENCES Transactions(transaction_id) ON DELETE CASCADE,
    product_id INT REFERENCES Products(product_id),
    quantity INT NOT NULL CHECK (quantity > 0),
    subtotal NUMERIC(10, 2) NOT NULL
);
INSERT INTO Users (username, password, role) VALUES
('superadmin', 'password123', 'superuser'),
('admin1', 'password123', 'admin'),
```

```sql
('user1', 'password123', 'user');


INSERT INTO Products (product_name, price, stock) VALUES

('Notebook', 5000, 100),

('Pen', 2000, 200),

('Eraser', 1500, 150);


INSERT INTO Transactions (cashier_name) VALUES

('Alice'),

('Bob');


INSERT INTO Transaction_Details (transaction_id, product_id, quantity, subtotal) VALUES

(1, 1, 2, 10000),

(1, 2, 5, 10000),

(2, 3, 3, 4500);



DO $$
BEGIN
  IF NOT EXISTS (SELECT FROM pg_roles WHERE rolname = 'superuser_role') THEN
    CREATE ROLE superuser_role;
  END IF;
  IF NOT EXISTS (SELECT FROM pg_roles WHERE rolname = 'admin_role') THEN
    CREATE ROLE admin_role;
  END IF;
  IF NOT EXISTS (SELECT FROM pg_roles WHERE rolname = 'user_role') THEN
    CREATE ROLE user_role;
  END IF;
END $$;


GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO superuser_role;
```

```sql
GRANT INSERT, UPDATE ON Products, Transactions, Transaction_Details TO admin_role;

GRANT SELECT ON ALL TABLES IN SCHEMA public TO user_role;

GRANT superuser_role TO superadmin;

GRANT admin_role TO admin1;

GRANT user_role TO user1;



SELECT
    t.transaction_id,
    t.transaction_date,
    t.cashier_name,
    p.product_name,
    td.quantity,
    td.subtotal,
    SUM(td.subtotal) OVER (PARTITION BY t.transaction_id) AS total
FROM
    Transactions t
JOIN Transaction_Details td ON t.transaction_id = td.transaction_id
JOIN Products p ON td.product_id = p.product_id
WHERE t.transaction_id = 1;
BEGIN;
UPDATE Products
SET stock = stock - 1
WHERE product_id = 1 AND stock >= 1;


INSERT INTO Transaction_Details (transaction_id, product_id, quantity, subtotal)
VALUES (1, 1, 1, 5000);
```

```
COMMIT;
```