Schema and table creation

```
-- Table for storing user details (Customer, Scheduler, Driver)
CREATE TABLE users (
   id INT AUTO_INCREMENT PRIMARY KEY,
   email VARCHAR(100) UNIQUE NOT NULL,
   password VARCHAR(255) NOT NULL,
   phone_number VARCHAR(15),
   role ENUM('customer', 'scheduler', 'driver') NOT NULL,
   truck_registration_number VARCHAR(50),
   truck_capacity INT
);

-- Table for storing products available for customers
CREATE TABLE products (
   id INT AUTO_INCREMENT PRIMARY KEY,
   name VARCHAR(255) NOT NULL,
   price DECIMAL(10, 2) NOT NULL
);

-- Table for storing deliveries
CREATE TABLE deliveries (
   id INT AUTO_INCREMENT PRIMARY KEY,
   user_id INT,
   product_id INT,
   quantity_kg DECIMAL(10, 2),
   delivery_date DATE,
   delivery_address VARCHAR(255),
   status ENUM('Pending', 'Completed') DEFAULT 'Pending',
   FOREIGN KEY(user_id) REFERENCES users(id),
   FOREIGN KEY(product_id) REFERENCES products(id)
```

```
);

-- Table for storing routes (used by schedulers)
CREATE TABLE routes (
  id INT AUTO_INCREMENT PRIMARY KEY,
  delivery_id INT,
  driver_id INT,
  route_date DATE,
  route_points TEXT, -- This could store a list of delivery addresses
  FOREIGN KEY(delivery_id) REFERENCES deliveries(id),
  FOREIGN KEY(driver_id) REFERENCES users(id)
);
```

Relationships And Associations

User

- One-to-Many with Delivery (user_id).
- One-to-Many with Route (driver_id for role "driver").

Product

- One-to-Many with Delivery (product_id).

Delivery

- Many-to-One with User (user_id).
- Many-to-One with Product (product_id).
- One-to-One with Route (delivery_id).

Route

- Many-to-One with User (driver role) (driver_id).
- One-to-One with Delivery (delivery_id).

- **User** ↔ **Delivery**: Tracks the customer making the delivery request.
- **Product** ↔ **Delivery**: Identifies the product being delivered.
- **Delivery** ↔ **Route**: Maps a delivery to its planned route.
- **User** ↔ **Route**: Assigns a route to a specific driver.

UML CLASS DIAGRAM FOR GOODS DELIVERY APPLICATION