**Database Relationship**

**Entities:**

* **Users**
  + id (primary key)
  + username
  + password
  + role
* **Products**
  + id (primary key)
  + name
  + description
  + price
  + is\_active
* **Subscriptions**
  + id (primary key)
  + user\_id (foreign key referencing Users)
  + product\_id (foreign key referencing Products)
  + frequency
  + start\_date
  + end\_date
  + is\_active
* **Orders**
  + id (primary key)
  + user\_id (foreign key referencing Users)
  + product\_id (foreign key referencing Products)
  + subscription\_id (foreign key referencing Subscriptions)
  + order\_date

**Relationships:**

* A user can have multiple subscriptions (one-to-many).
* A subscription is associated with one user (many-to-one).
* A product can be part of multiple subscriptions (one-to-many).
* An order is associated with one user (one-to-one).
* An order is associated with many products (many-to-one).

Table Creation Queries

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) UNIQUE NOT NULL,

password VARCHAR(100) NOT NULL,

role ENUM('ADMIN', 'CUSTOMER') NOT NULL

);

CREATE TABLE products (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

description TEXT,

price DECIMAL(10, 2) NOT NULL,

is\_active BOOLEAN DEFAULT TRUE

);

CREATE TABLE subscriptions (

id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

product\_id INT,

frequency ENUM('WEEKLY', 'BI-WEEKLY', 'MONTHLY'),

start\_date DATE,

end\_date DATE,

is\_active BOOLEAN DEFAULT TRUE,

FOREIGN KEY (user\_id) REFERENCES users(id),

FOREIGN KEY (product\_id) REFERENCES products(id)

);

CREATE TABLE orders (

id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

product\_id INT,

subscription\_id INT,

order\_date DATE,

FOREIGN KEY (user\_id) REFERENCES users(id),

FOREIGN KEY (product\_id) REFERENCES products(id),

FOREIGN KEY (subscription\_id) REFERENCES subscriptions(id)

);