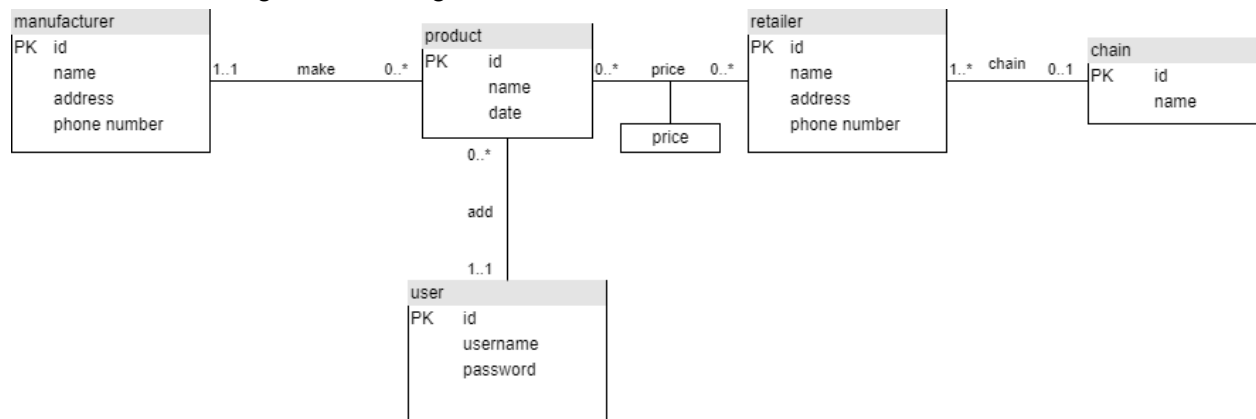


This is the UML diagram we designed for our database



Overview: The main entity in our application is the product. Each product can be made only by one manufacturer (one-many). Many retailers can offer different products and at different prices (many-many). A retailer can be a part of a chain (like Target, Walmart which have many stores at different locations and can sometimes offer different prices). Other than product information that is scraped from the internet by bot, a user can also add a product if it is not on the database.

#### Description of Entities:

Manufacturer - The company that makes products

Product - The products made by manufacturer

Retailer - Stores that sell products

User - User of the produce application

Chain - List of popular chain retailers

#### Descriptions of Relationships:

Make - Manufacturers can make one or more products but products only have one manufacturer

Price - Products are priced by retailers, retailers price products

Chain - Retailers can be apart of at most one chain, Each chain can have many retailers

Add - Users can add 0 or more products, Each product can only be added by one user.

#### Relational Schema:

Manufacturer(id: INT [PK], name: VAR(255), address: VAR(255), phone number: INT)

Product(id: INT [PK], name: VAR(255), date: VAR(255))

Retailer(id: INT [PK], name: VAR(255) , address: VAR(255), phone number: INT)

User(username: VAR(255) [PK], password: VAR(255))

Chain(id: INT [PK], name: VAR(255))

Add(product.id: INT, user.id: INT)

Make(manufacturer.id: INT, product.id: INT)

Price(product.id: INT, retailer.id: INT, price: INT)

Chain(retailer.id: INT, chain.id: INT)