DBMS Mini Project

Synopsis

Grocery Store Management System

Problem Statement:

To implement Database Management System for effectively managing activities at the Grocery Store.

Objectives:

Maintaining the records in grocery shop manually is a tedious task. The main objective of the Grocery Store Management System is to automate the existing manual system of maintaining the records. Grocery Store Management System can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather than to concentrate on the record keeping. This system will help in maintaining computerized records without redundant entries. This means that one need not be distracted by data that is not relevant, while being able to access the data.

Functional requirements:

The system consists of two main modules:

1)For Managing Products

2)For Managing Orders

i)In module related to Managing Products functionality is provided for adding new products to the database by providing its name, unit of measurement and price per unit.

ii)Similarly there is an option for deleting an existing product from the database.

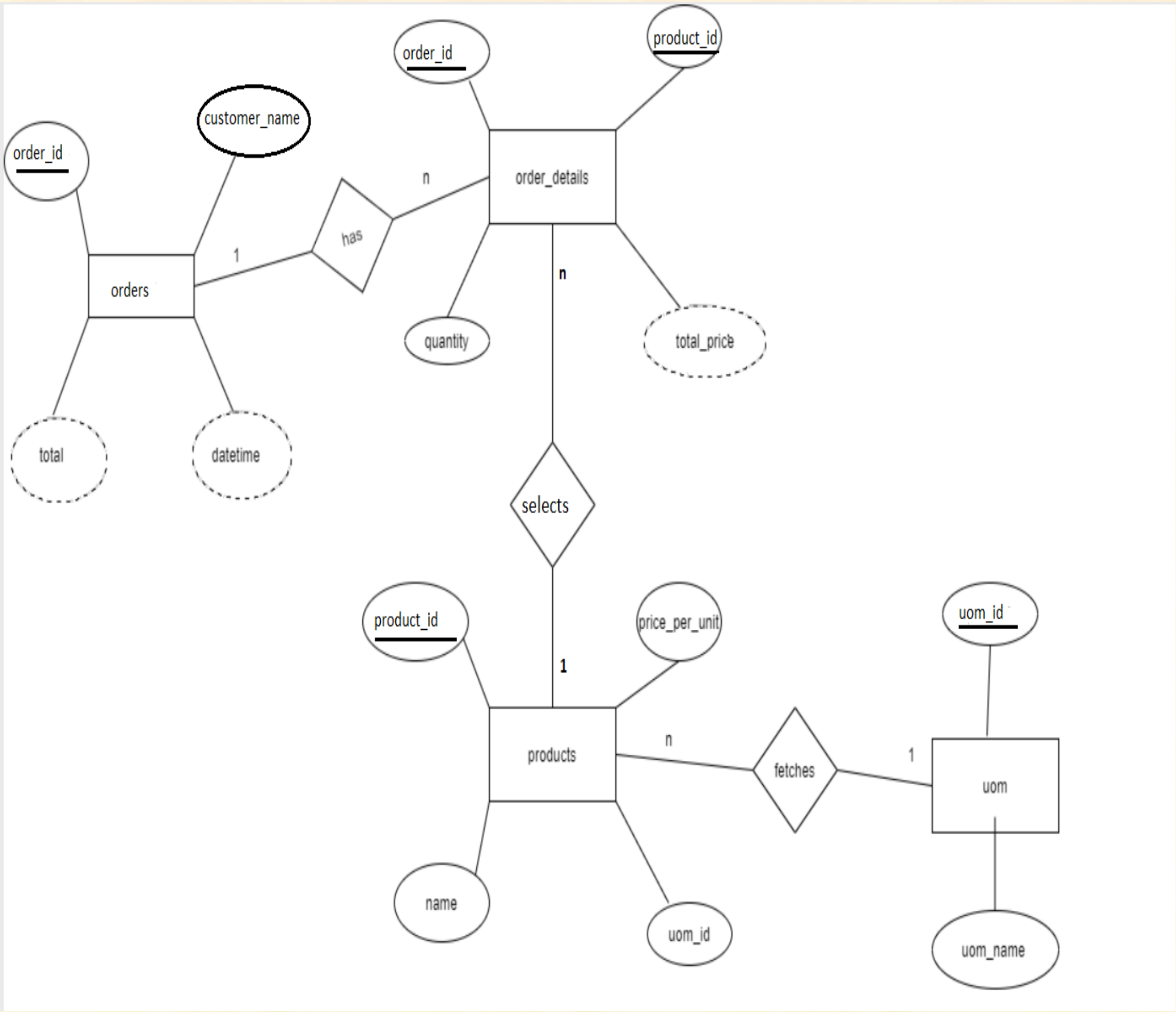
iii)In module related to Managing Orders functionality is provided for adding an item in an order by selecting its name from the products list and selecting required number of units. Total price for the selected item is calculated automatically by multiplication of number of units selected and price per unit.

iv)Option for deleting an item is also provided.

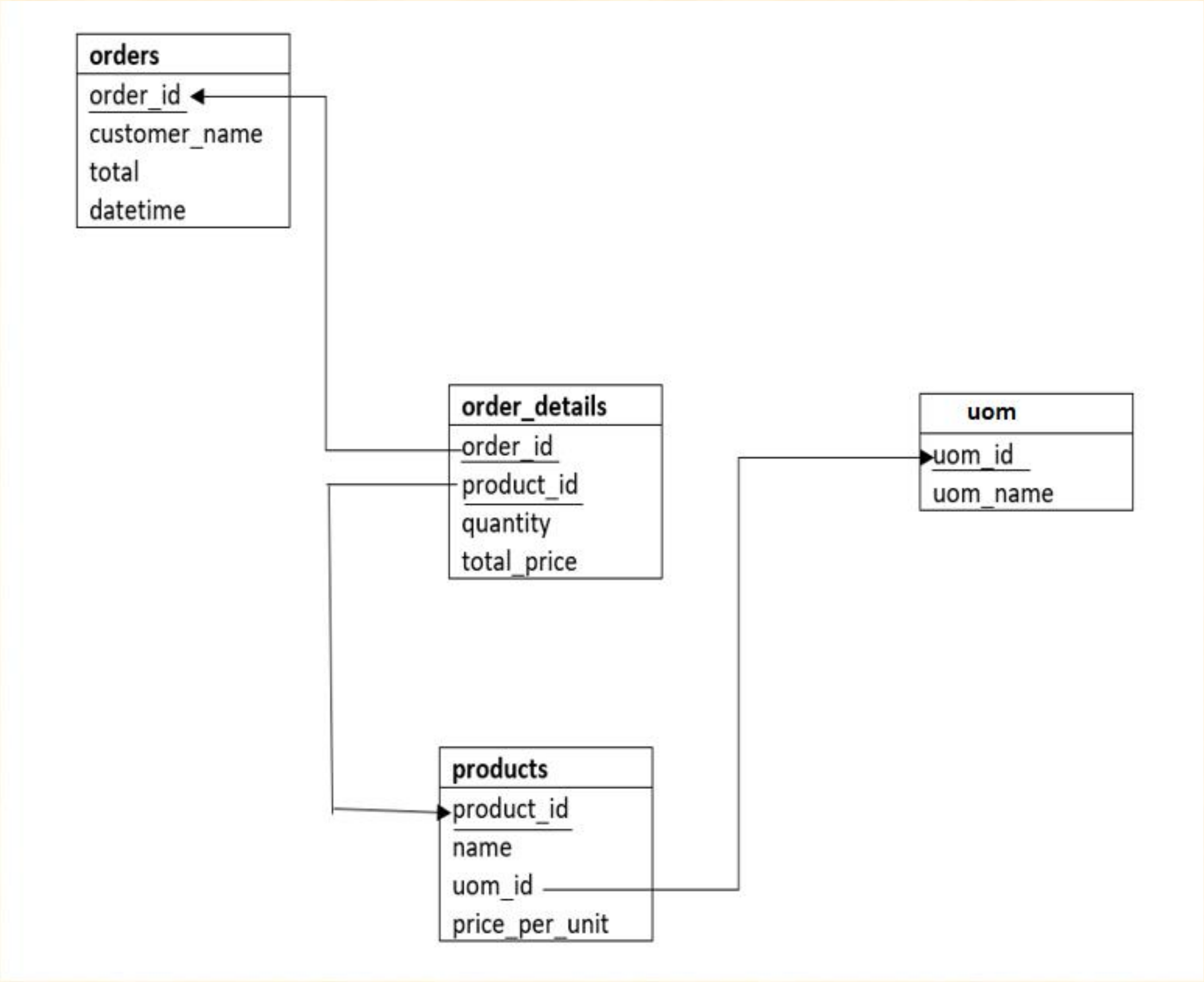
v)Also details of an order can be viewed.

1. **Front end**: UI is written in HTML / CSS / Javascript / Bootstrap
2. **Backend**: Python and Flask
3. **Database**: MYSQL

ER diagram:


order_id
customer_name
orders
1
has
n
order_id
product_id
order_details
n
quantity
total_price
total
datetime
selects
product_id
price_per_unit
uom_id
1
products
n
fetches
1
uom
name
uom_id
uom_name

Relational schemas obtained from ER Diagram:



Set of Functional dependencies that must hold on each table:

- (products) product_id -> (products) name, uom_id, price_per_unit
- (orders) order_id -> (orders) customer_name, total, datetime
- (uom) uom_id -> (uom) uom_name
- (order_details) order_id, product_id -> (order_details) quantity, total_price