*products(p\_id, name, price, rating)*

*stores(s\_id, l\_id, name, business\_hours, phone\_number)*

*sells(p\_id, s\_id)*

*location(l\_id, street\_name, street\_num, city, state, zip)*

*customer(c\_id, name, street\_name, street\_num, city, state, zip)*

*c\_phone\_number(c\_id, phone\_number)*

*premium\_membership(reward\_num, c\_id, discount\_amt, date\_joined)*

*order(o\_num, c\_id, o\_date, o\_total)*

*delivers(p\_id, o\_number, quantity, date\_shipped, expected\_delivery)*

1. Products
   1. View product information?
   2. Change rating of product?
   3. Add new product?
   4. Remove product?
2. Stores (was initially gonna search by state and present all stores in area, but proved to be too complex)
   1. View store information? [store w/ location]
3. Sells (attempting to display both name and price, especially of store carried more than one item. Displayed for every item carried, fixed by fixing my query)
   1. Search for whole catalog listing from specific store? [sells, products]
   2. Find particular item? [sells, products]
   3. Add new product to catalog?
   4. Remove product from catalog?
4. Customer [did not find it necessary to delete customer since they’d only be registered upon making a single purchase]
   1. View customer information?
   2. Update customer information?
   3. Register new customer? (figuring out how to account for secondary phone numbers)
5. Premium membership
   1. Register customer for membership?
   2. Cancel membership for particular customer
6. Order/Delivery
   1. Add new order
   2. Cancel order [from delivery]
   3. View delivery status for particular order?
7. Update information in a particular table
   1. Change rating of product
   2. Update status of store
   3. Update sales catalog
   4. Update location
   5. Update customer information
   6. Update customer’s premium status
   7. Update order
   8. Update delivery status
8. Remove information
   1. Remove stock of a product
   2. Remove store information
   3. Remove item from catalog
   4. Remove store location
   5. Remove customer information
   6. Cancel customer’s premium status
   7. Cancel order
   8. Cancel pending delivery
9. Insert information
   1. Register new product
   2. Register new store
   3. Add new item to catalog
   4. Add new store location
   5. Register new customer
   6. Register customer for premium membership
   7. Add new order
   8. Add new delivery request
10. View delivery status for particular order
    1. Join order, delivers, customer
    2. Display customer name/id, order number/date/total, part id, date shipped/expected delivery
11. View premium status for particular customer
    1. Join customer, premium membership
    2. Display customer name/id, reward num, date\_joined, discount amount
12. View information about specific store
    1. Join store, location
    2. Display s\_id, address, business hrs, phone number
13. View premium status for particular customer
    1. Join customer, premium membership
    2. Display customer name/id, reward num, date\_joined, discount amount

#2. Add New Order

def newOrder():

oNum = input("Please enter order number: ")

cId = input("Please enter customer id: ")

query1 = "select \* from orders where o\_number = \"" + oNum + "\";"

query2 = "select \* from customer where c\_id = \"" + cId + "\";"

mycursor.execute(query1)

data1 = mycursor.fetchall()

mycursor.execute(query2)

data2 = mycursor.fetchall()

#Print data

if data1:

print("Order already exists.")

return

elif not data2:

print("Customer does not exist.")

return

else:

#Insert records for new order

oDate = input("Enter date ordered: ")

oTotal = input("Enter order total: ")

#check to see if customer gets discount

rewQuery = "select \* from premium\_membership where c\_id = " + cId + ";"

mycursor.execute(rewQuery)

rewData = mycursor.fetchall()

if rewData:

oQuery = "insert into orders values (\"" + oNum + "\", " + cId + ", \"" + oDate + "\", " + oTotal + "-(.1\*" + oTotal + "));"

else:

oQuery = "insert into orders values (" + oNum + ", " + cId + ", \"" + oDate + "\", \"" + oTotal + "\");"

mycursor.execute(oQuery)

conn.commit()

print("Order processed in system.")

return