

PROGRAM:

```
import sqlite3
from tkinter import *
from tkinter import ttk

root = Tk()
root.title("Inventory System")
root.geometry("1080x720")
my_tree = ttk.Treeview(root)
storeName = "Inventory System"

def reverse(tuples):
    new_tup = tuples[::-1]
    return new_tup

def insert( id, name, price, quantity):
    conn = sqlite3.connect("data.db")
    cursor = conn.cursor()

    cursor.execute("""CREATE TABLE IF NOT EXISTS
inventory(itemId TEXT, itemName TEXT, itemPrice TEXT,
itemQuantity TEXT)""")

    cursor.execute("INSERT INTO inventory VALUES ('" + str(id) +
"', '" + str(name) + "', '" + str(price) + "', '" + str(quantity) +
"')" )
    conn.commit()

def delete(data):
    conn = sqlite3.connect("data.db")
    cursor = conn.cursor()

    cursor.execute("""CREATE TABLE IF NOT EXISTS
inventory(itemId TEXT, itemName TEXT, itemPrice TEXT,
itemQuantity TEXT)""")

    cursor.execute("DELETE FROM inventory WHERE itemId = '" +
str(data) + "'")
```

```

conn.commit()

def update(id, name, price, quantity, idName):
    conn = sqlite3.connect("data.db")
    cursor = conn.cursor()

    cursor.execute("""CREATE TABLE IF NOT EXISTS
        inventory(itemId TEXT, itemName TEXT, itemPrice TEXT,
itemQuantity TEXT)""")

    cursor.execute("UPDATE inventory SET itemId = '" + str(id) + "',
itemName = '" + str(name) + "', itemPrice = '" + str(price) + "',
itemQuantity = '" + str(quantity) + "' WHERE
itemId='"+str(idName)+"'")
    conn.commit()

def read():
    conn = sqlite3.connect("data.db")
    cursor = conn.cursor()

    cursor.execute("""CREATE TABLE IF NOT EXISTS
        inventory(itemId TEXT, itemName TEXT, itemPrice TEXT,
itemQuantity TEXT)""")

    cursor.execute("SELECT * FROM inventory")
    results = cursor.fetchall()
    conn.commit()
    return results

def insert_data():
    itemId = str(entryId.get())
    itemName = str(entryName.get())
    itemPrice = str(entryPrice.get())
    itemQuantity = str(entryQuantity.get())
    if itemId == "" or itemName == " ":
        print("Error Inserting Id")
    if itemName == "" or itemName == " ":
        print("Error Inserting Name")
    if itemPrice == "" or itemPrice == " ":

```

```

        print("Error Inserting Price")
    if itemQuantity == "" or itemQuantity == " ":
        print("Error Inserting Quantity")
    else:
        insert(str(itemId), str(itemName), str(itemPrice),
str(itemQuantity))

    for data in my_tree.get_children():
        my_tree.delete(data)

    for result in reverse(read()):
        my_tree.insert(parent='', index='end', iid=result, text="",
values=(result), tag="orow")

    my_tree.tag_configure('orow', background='#EEEEEE')
    my_tree.grid(row=1, column=5, columnspan=4, rowspan=5, padx=10,
pady=10)

def delete_data():
    selected_item = my_tree.selection()[0]
    deleteData = str(my_tree.item(selected_item)['values'][0])
    delete(deleteData)

    for data in my_tree.get_children():
        my_tree.delete(data)

    for result in reverse(read()):
        my_tree.insert(parent='', index='end', iid=result, text="",
values=(result), tag="orow")

    my_tree.tag_configure('orow', background='#EEEEEE')
    my_tree.grid(row=1, column=5, columnspan=4, rowspan=5, padx=10,
pady=10)

def update_data():
    selected_item = my_tree.selection()[0]
    update_name = my_tree.item(selected_item)['values'][0]
    update(entryId.get(), entryName.get(), entryPrice.get(),
entryQuantity.get(), update_name)

    for data in my_tree.get_children():

```

```

my_tree.delete(data)

for result in reverse(read()):
    my_tree.insert(parent='', index='end', iid=result, text="",
values=(result), tag="orow")

my_tree.tag_configure('orow', background='#EEEEEE')
my_tree.grid(row=1, column=5, columnspan=4, rowspan=5, padx=10,
pady=10)

titleLabel = Label(root, text=storeName, font=('Arial bold', 30),
bd=2)
titleLabel.grid(row=0, column=0, columnspan=8, padx=20, pady=20)

idLabel = Label(root, text="ID", font=('Arial bold', 15))
nameLabel = Label(root, text="Name", font=('Arial bold', 15))
priceLabel = Label(root, text="Price", font=('Arial bold', 15))
quantityLabel = Label(root, text="Quantity", font=('Arial bold',
15))
idLabel.grid(row=1, column=0, padx=10, pady=10)
nameLabel.grid(row=2, column=0, padx=10, pady=10)
priceLabel.grid(row=3, column=0, padx=10, pady=10)
quantityLabel.grid(row=4, column=0, padx=10, pady=10)

entryId = Entry(root, width=25, bd=5, font=('Arial bold', 15))
entryName = Entry(root, width=25, bd=5, font=('Arial bold', 15))
entryPrice = Entry(root, width=25, bd=5, font=('Arial bold', 15))
entryQuantity = Entry(root, width=25, bd=5, font=('Arial bold', 15))
entryId.grid(row=1, column=1, columnspan=3, padx=5, pady=5)
entryName.grid(row=2, column=1, columnspan=3, padx=5, pady=5)
entryPrice.grid(row=3, column=1, columnspan=3, padx=5, pady=5)
entryQuantity.grid(row=4, column=1, columnspan=3, padx=5, pady=5)

buttonEnter = Button(
    root, text="Enter", padx=5, pady=5, width=5,
    bd=3, font=('Arial', 15), bg="#0099ff", command=insert_data)
buttonEnter.grid(row=5, column=1, columnspan=1)

buttonUpdate = Button(
    root, text="Update", padx=5, pady=5, width=5,
    bd=3, font=('Arial', 15), bg="#ffff00", command=update_data)

```

```

buttonUpdate.grid(row=5, column=2, columnspan=1)

buttonDelete = Button(
    root, text="Delete", padx=5, pady=5, width=5,
    bd=3, font=('Arial', 15), bg="#e62e00", command=delete_data)
buttonDelete.grid(row=5, column=3, columnspan=1)

style = ttk.Style()
style.configure("Treeview.Heading", font=('Arial bold', 15))

my_tree['columns'] = ("ID", "Name", "Price", "Quantity")
my_tree.column("#0", width=0, stretch=NO)
my_tree.column("ID", anchor=W, width=100)
my_tree.column("Name", anchor=W, width=200)
my_tree.column("Price", anchor=W, width=150)
my_tree.column("Quantity", anchor=W, width=150)
my_tree.heading("ID", text="ID", anchor=W)
my_tree.heading("Name", text="Name", anchor=W)
my_tree.heading("Price", text="Price", anchor=W)
my_tree.heading("Quantity", text="Quantity", anchor=W)

for data in my_tree.get_children():
    my_tree.delete(data)

for result in reverse(read()):
    my_tree.insert(parent='', index='end', iid=0, text="",
values=(result), tag="orow")

my_tree.tag_configure('orow', background='#EEEEEE', font=('Arial
bold', 15))
my_tree.grid(row=1, column=5, columnspan=4, rowspan=5, padx=10,
pady=10)

root.mainloop()

```

OUTPUT:

Inventory System

Inventory System

ID

Name

Price

Quantity

Enter

Update

Delete

ID	Name	Price	Quantity
----	------	-------	----------

Inventory System

Inventory System

ID

86594

Name

WATERBOTTLE

Price

500

Quantity

4

Enter

Update

Delete

ID	Name	Price	Quantity
----	------	-------	----------

Inventory System

Inventory System

ID

86594

Name

WATERBOTTLE

Price

500

Quantity

4

Enter

Update

Delete

ID	Name	Price	Quantity
86594	WATERBOTTLE	500	4

Inventory System

Inventory System

ID

5648974

Name

KEYCHAIN

Price

150

Quantity

7

Enter

Update

Delete

ID	Name	Price	Quantity
5648974	KEYCHAIN	150	7
86594	WATERBOTTLE	500	4

Inventory System

Inventory System

ID

5648974

Name

KEYCHAIN

Price

150

Quantity

15

Enter

Update

Delete

ID	Name	Price	Quantity
5648974	KEYCHAIN	150	15
86594	WATERBOTTLE	500	4

Inventory System

Inventory System

ID

Name

Price

Quantity

Enter

Update

Delete

ID	Name	Price	Quantity
5648974	KEYCHAIN	150	15
86594	WATERBOTTLE	500	4

Inventory System

Inventory System

ID

Name

Price

Quantity

Enter

Update

Delete

ID	Name	Price	Quantity
86594	WATERBOTTLE	500	4