

ITEM.java

```
package org.grocery.stock;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "ITEM_TBL")
public class Item {
    private static int totalNumberOfProducts;

    static {
        Item.totalNumberOfProducts=0;
    }

    public static int getTotalNumberOfProducts() {
        return totalNumberOfProducts;
    }

    @GeneratedValue
    @Id
    @Column(name = "ITEM_ID")
    private int id;
    @Column(name = "NAME")
    private String name;
    @Column(name = "QTY")
    private int quantity;
    @Column(name = "PRICE")
    private double price;

    public Item() {
        super();
        Item.totalNumberOfProducts++;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
```

```

        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getQuantity() {
        return quantity;
    }

    public void setQuantity(int quantity) {
        this.quantity = quantity;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }
}

```

Solution.java

```

package org.main;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

import org.grocery.stock.Item;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class Solution {
    public static void main(String[] args) throws IOException {
        SessionFactory sf = new Configuration().configure().buildSessionFactory();
        Session session = sf.openSession();
        int vid;
        BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));

        while(true) {

```

```

System.out.println("Enter your choice:");
int ch = Integer.valueOf(bf.readLine());
session.beginTransaction();
switch (ch) {
case 1:

    Item item = new Item();
    System.out.println("Enter a name:");
    String name = bf.readLine();
    System.out.println("Enter a price:");
    double price = Double.valueOf(bf.readLine());
    System.out.println("Enter a quantity:");
    int qty = Integer.valueOf(bf.readLine());
    System.out.println("one record inserted");
    item.setName(name);
    item.setPrice(price);
    item.setQuantity(qty);
    session.save(item);
    session.getTransaction().commit();
    break;

case 2:

    System.out.println("Enter a valid id to update");
    vid = Integer.valueOf(bf.readLine());
    Item item1 = session.get(Item.class, vid);
    item1.setName("subasri");
    item1.setPrice(1542784.254);
    session.update(item1);
    session.getTransaction().commit();
    System.out.println("Record updated");
    break;

case 3:

    Item item2 = new Item();
    item2.setId(1);
    session.delete(item2);
    session.getTransaction().commit();
    System.out.println("Record deleted");
    session.getTransaction().commit();
    break;

case 4:

    System.out.println("Enter a valid id to search");
    vid = Integer.valueOf(bf.readLine());
    Item item3 = session.get(Item.class, vid);
    System.out.println(item3.getId());
    System.out.println(item3.getName());
    System.out.println(item3.getPrice());

```

```
        System.out.println(item3.getQuantity());
        session.getTransaction().commit();
        break;
    case 5:
        System.out.println("total no of products="+Item.getTotalNumberOfProducts());
        session.getTransaction().commit();
        break;
    case 6:
        System.out.println("Exit");
        System.exit(0);
        break;
    default:
        break;
}

session.close();
sf.close();
}
}
```