

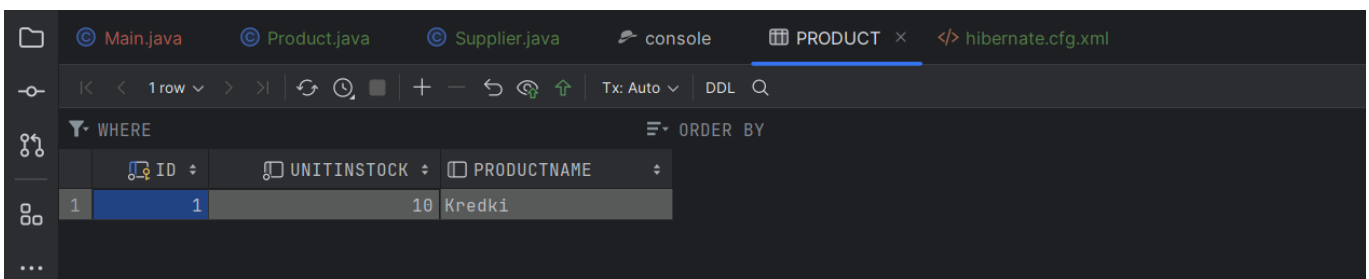
```
package org.example;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;

4 usages new *
@Entity
public class Product {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private int id;
    1 usage
    private String productName;
    1 usage
    private int unitInStock;

    new *
    public Product(){

    }
    1 usage new *
    public Product(String productName, int unitsInStock){
        this.productName = productName;
        this.unitInStock = unitsInStock;
    }
}
```



ID	UNITINSTOCK	PRODUCTNAME
1	10	Kredki

```
public class Main {  
  
    5 usages  
    private static SessionFactory sessionFactory = null;  
  
    public static void main(String[] args) {  
        sessionFactory = getSessionFactory();  
        Session session = sessionFactory.openSession();  
        Product product = new Product( productName: "Kredki", unitsInStock: 10);  
        Transaction tx = session.beginTransaction();  
        session.persist(product);  
        tx.commit();  
        session.close();  
    }  
  
    1 usage  
    private static SessionFactory getSessionFactory(){  
        if (sessionFactory == null){  
            Configuration configuration = new Configuration();  
            sessionFactory = configuration.configure().buildSessionFactory();  
        }  
        return sessionFactory;  
    }  
}
```



```

package org.example;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;

6 usages new *
@Entity
public class Supplier {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private int id;
    1 usage
    private String companyName;
    1 usage
    private String street;
    1 usage
    private String city;

    1 usage new *
    public Supplier(String companyName, String street, String city){
        this.companyName = companyName;
        this.street = street;
        this.city = city;
    }
    new *
    public Supplier() {}
}

```

```

1 package org.example;
2
3 import jakarta.persistence.*;
4
5 4 usages new *
6 @Entity
7 public class Product {
8     @Id
9     @GeneratedValue(strategy = GenerationType.AUTO)
10    private int id;
11    1 usage
12    private String productName;

```

```
10 private String productName;
    1 usage
11 private int unitInStock;
    1 usage
12 @ManyToOne
13 private Supplier supplier;
    new *
14 public Product(){}
    1 usage new *
15 public Product(String productName, int unitsInStock){
16     this.productName = productName;
17     this.unitInStock = unitsInStock;
18 }
19
    1 usage new *
20 public void setSupplier(Supplier supplier) {
21     this.supplier = supplier;
22 }
23 }
```

```
public class Main {

    5 usages
    private static SessionFactory sessionFactory = null;

    public static void main(String[] args) {
        sessionFactory = getSessionFactory();
        Session session = sessionFactory.openSession();
        Supplier supplier = new Supplier(CompanyName: "Ikea1", Street: "Main Street", City: "New York");
        Product product = new Product(productName: "Zabawki1", unitsInStock: 10);
        product.setSupplier(supplier);
        Transaction tx = session.beginTransaction();
        session.persist(supplier);
        session.persist(product);
        tx.commit();
        session.close();
    }

    1 usage
    private static SessionFactory getSessionFactory(){
        if (sessionFactory == null){
            Configuration configuration = new Configuration();
            sessionFactory = configuration.configure().buildSessionFactory();
        }
        return sessionFactory;
    }
}
```

	ID	SUPPLIER_ID	UNITINSTOCK	PRODUCTNAME
1	1	1	10	Zabawki
2	2	2	10	Zabawki1

	ID	CITY	COMPANYNAME	STREET
1	1	New York	Ikea	Main Street

```
public class Main {  
  
    5 usages  
    private static SessionFactory sessionFactory = null;  
  
    public static void main(String[] args) {  
        sessionFactory = getSessionFactory();  
        Session session = sessionFactory.openSession();  
        Supplier supplier = new Supplier( CompanyName: "Ikea2", Street: "Main Street", City: "New York");  
        Product product = session.createQuery( s: "FROM Product p ORDER BY p.id", Product.class)  
            .setMaxResults(1)  
            .uniqueResult();  
        product.setSupplier(supplier);  
        Transaction tx = session.beginTransaction();  
        session.persist(supplier);  
        session.persist(product);  
        tx.commit();  
        session.close();  
    }  
  
    1 usage  
    private static SessionFactory getSessionFactory(){  
        if (sessionFactory == null){  
            Configuration configuration = new Configuration();  
            sessionFactory = configuration.configure().buildSessionFactory();  
        }  
        return sessionFactory;  
    }  
}
```

	ID	SUPPLIER_ID	UNITINSTOCK	PRODUCTNAME
1	1	52	10	Zabawki
2	2	2	10	Zabawki1

	ID	CITY	COMPANYNAME	STREET
1	1	New York	Ikea	Main Street
2	2	New York	Ikea1	Main Street
3	52	New York	Ikea2	Main Street