

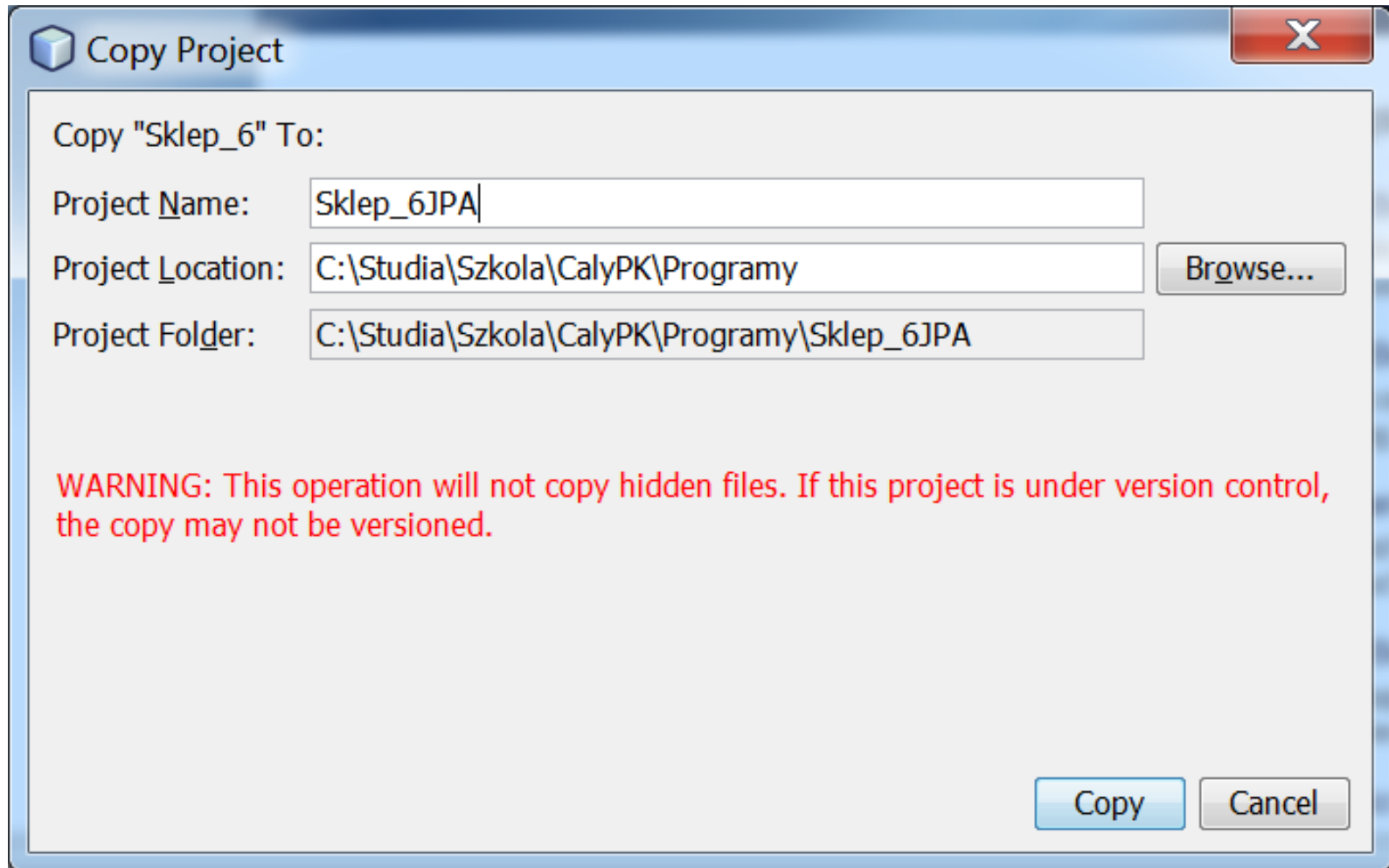
# **Podstawowe informacje o technologii Java Persistence API - przykład**

na podstawie

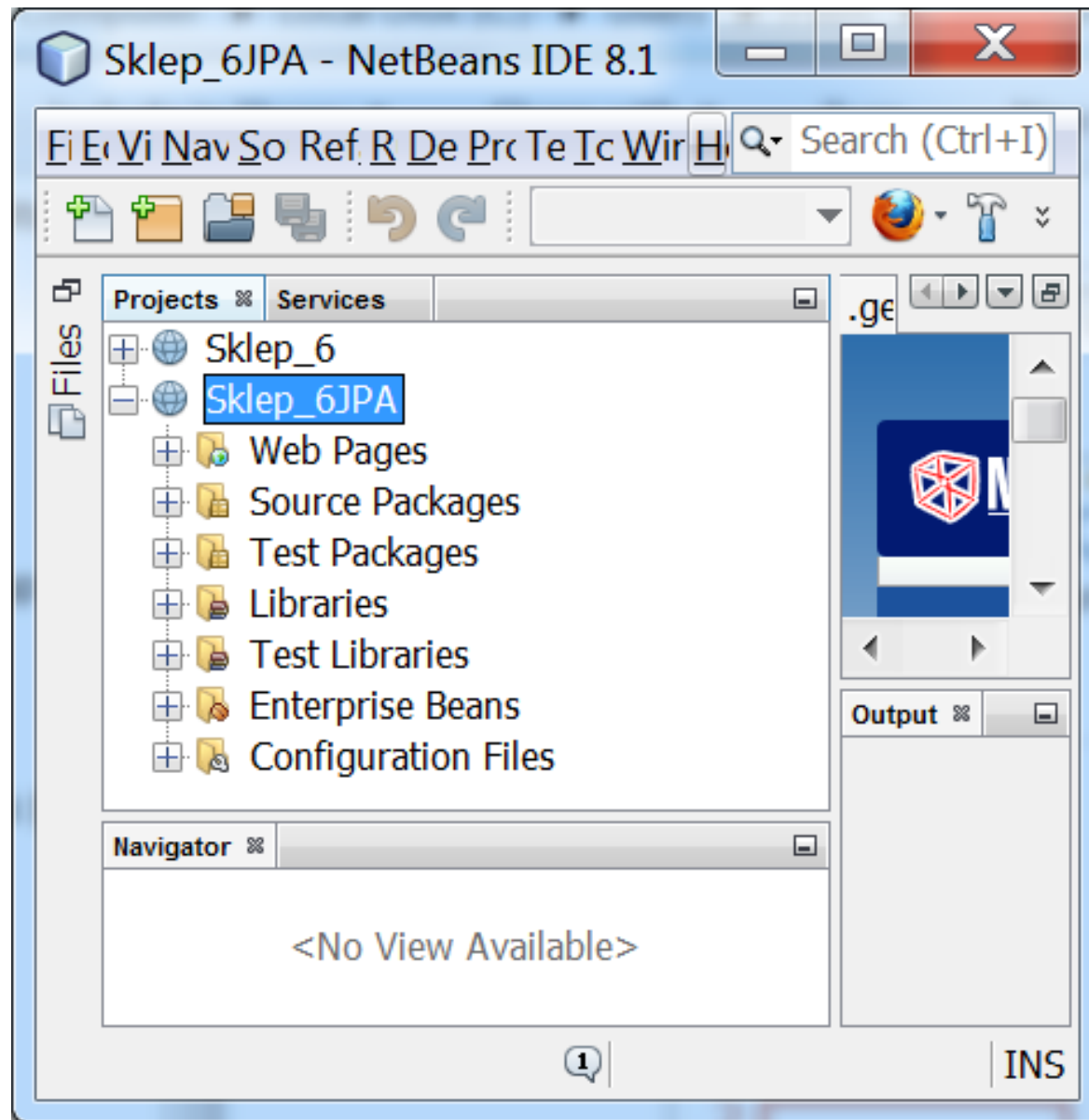
<https://docs.oracle.com/javaee/7/JEETT.pdf>

## **Programowanie komponentowe 3\_2**

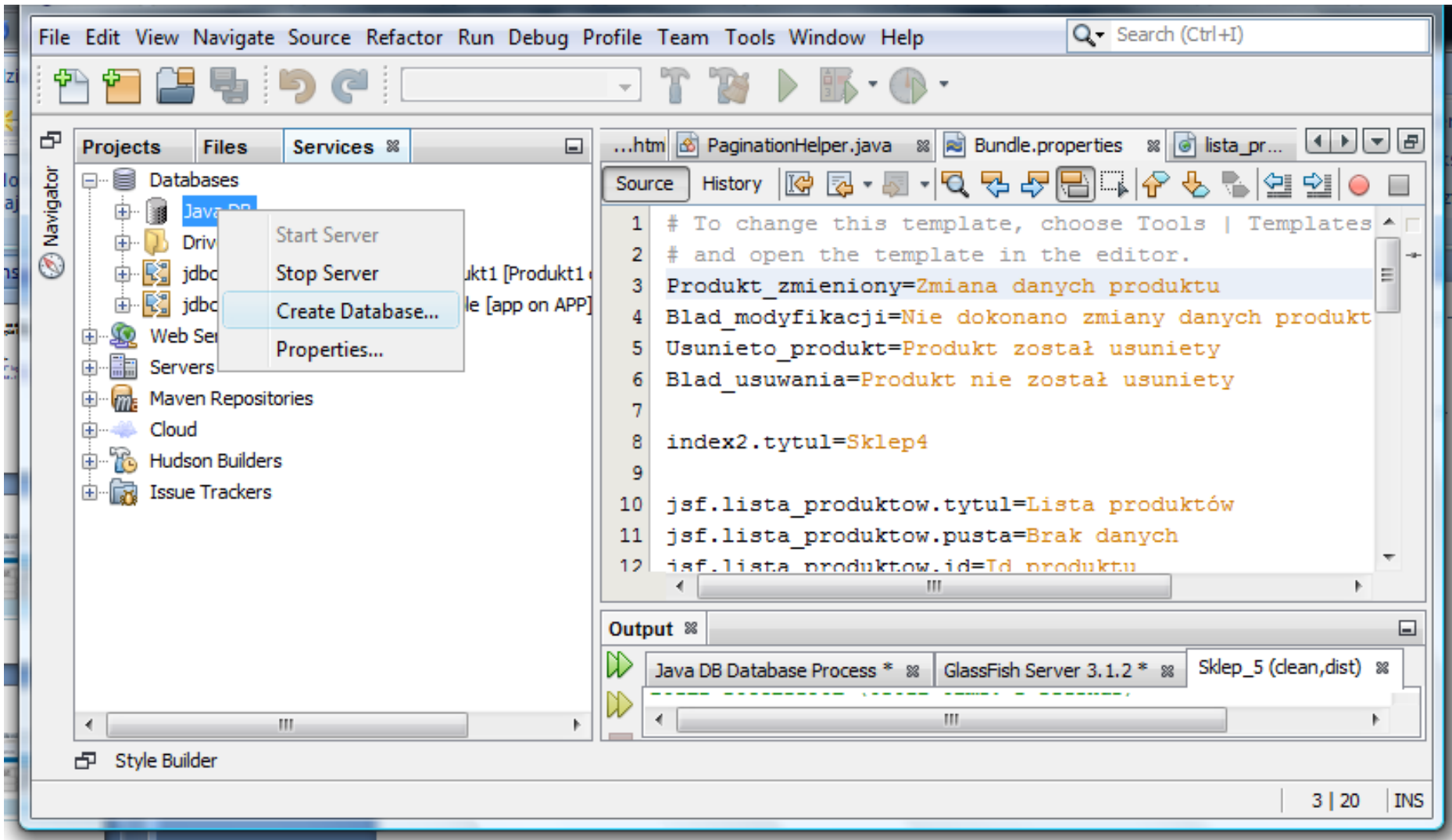
# 1. Wykonanie kopii programu Sklep\_6



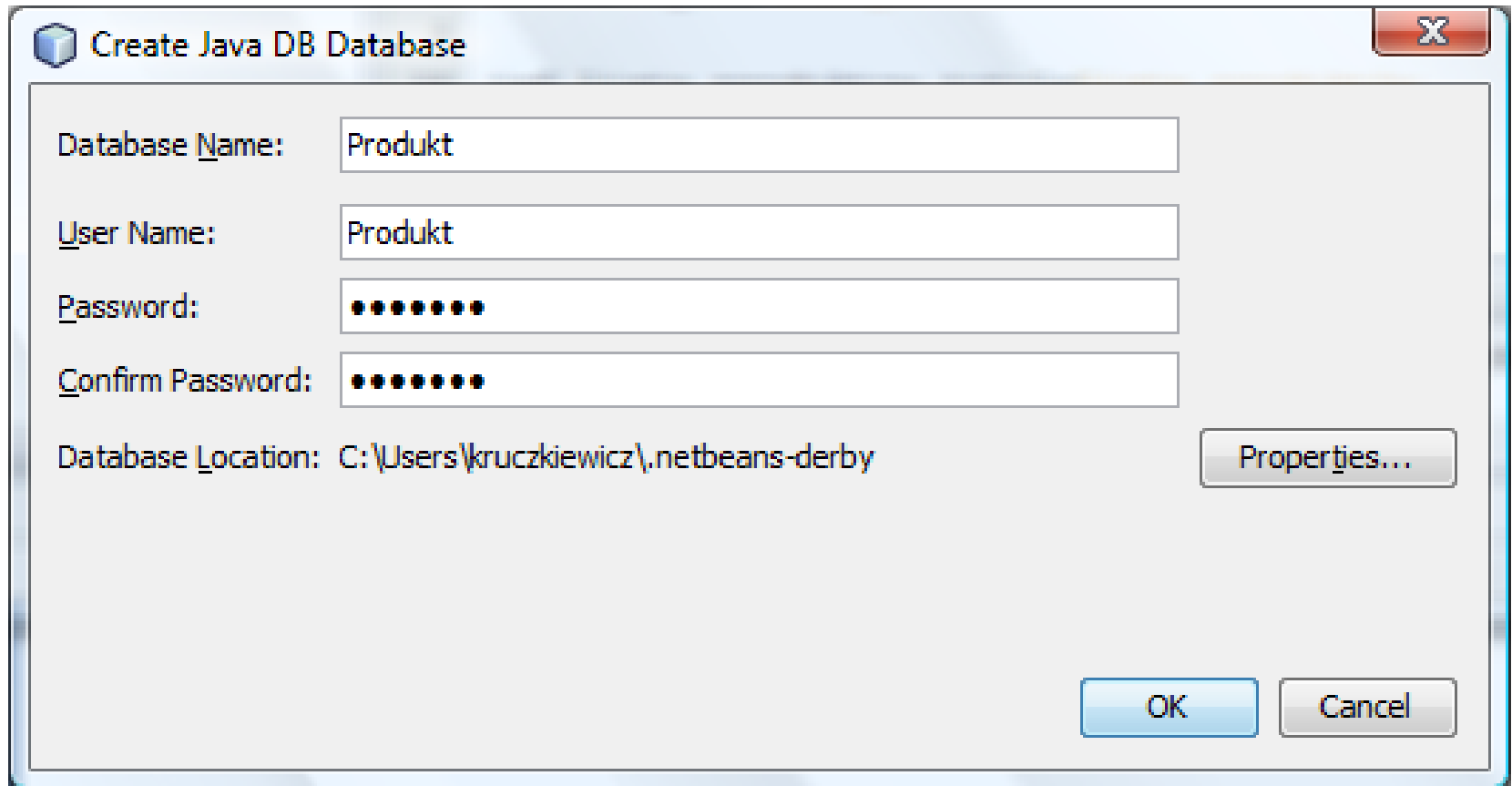
# 1. Wykonanie kopii programu Sklep\_6 (cd)



2. Należy utworzyć pustą bazę danych w zakładce **Services** (prawy przycisk myszy na pozycji JavaDB po rozwinięciu pozycji Database i wybór **Create Database**)



### 3. Wpisanie nazwy bazy danych (Database Name), loginu (User Name) i hasła (Password)



Create Java DB Database

Database Name: Produkt

User Name: Produkt

Password: ●●●●●●

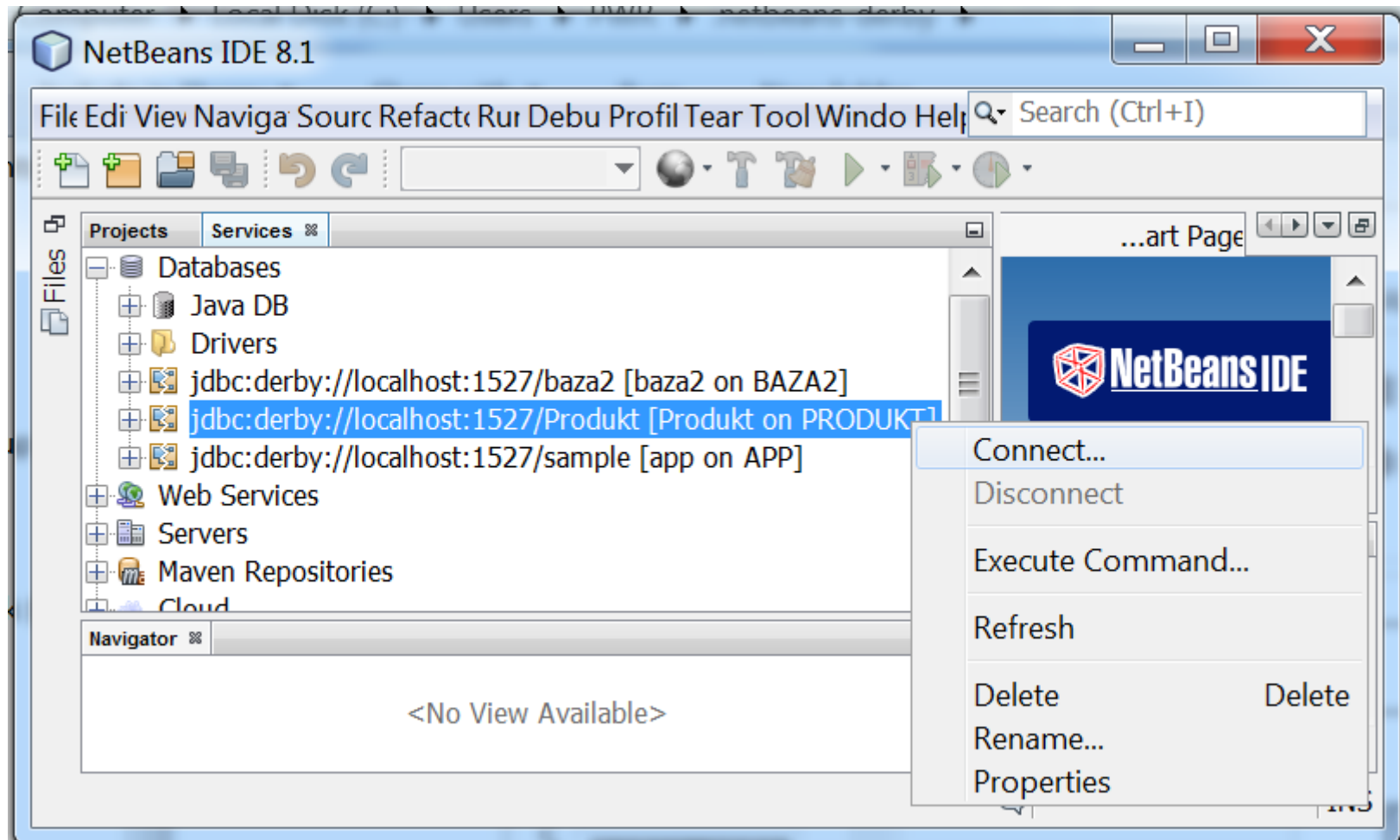
Confirm Password: ●●●●●●

Database Location: C:\Users\kruczkiewicz\.netbeans-derby

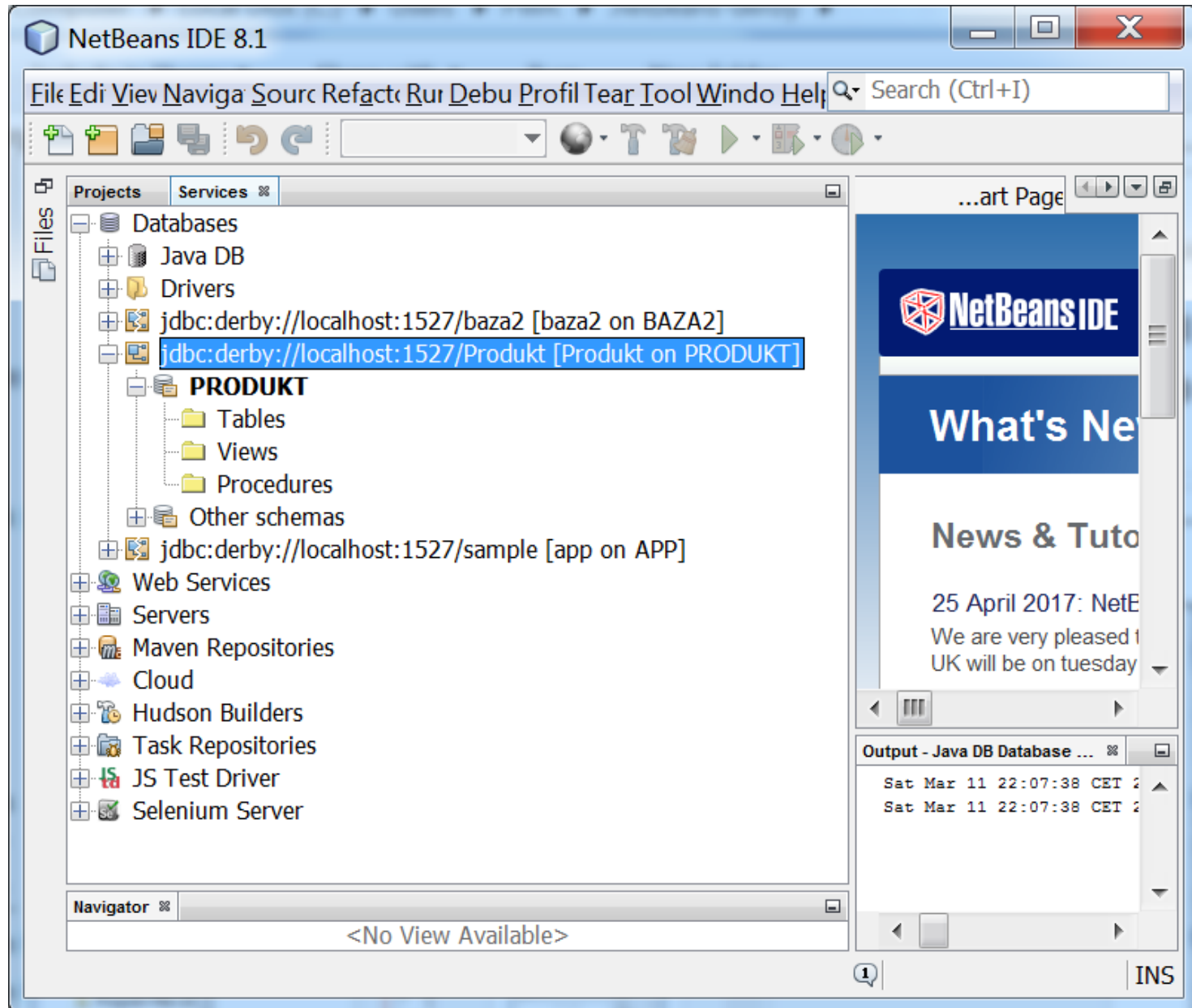
Properties...

OK Cancel

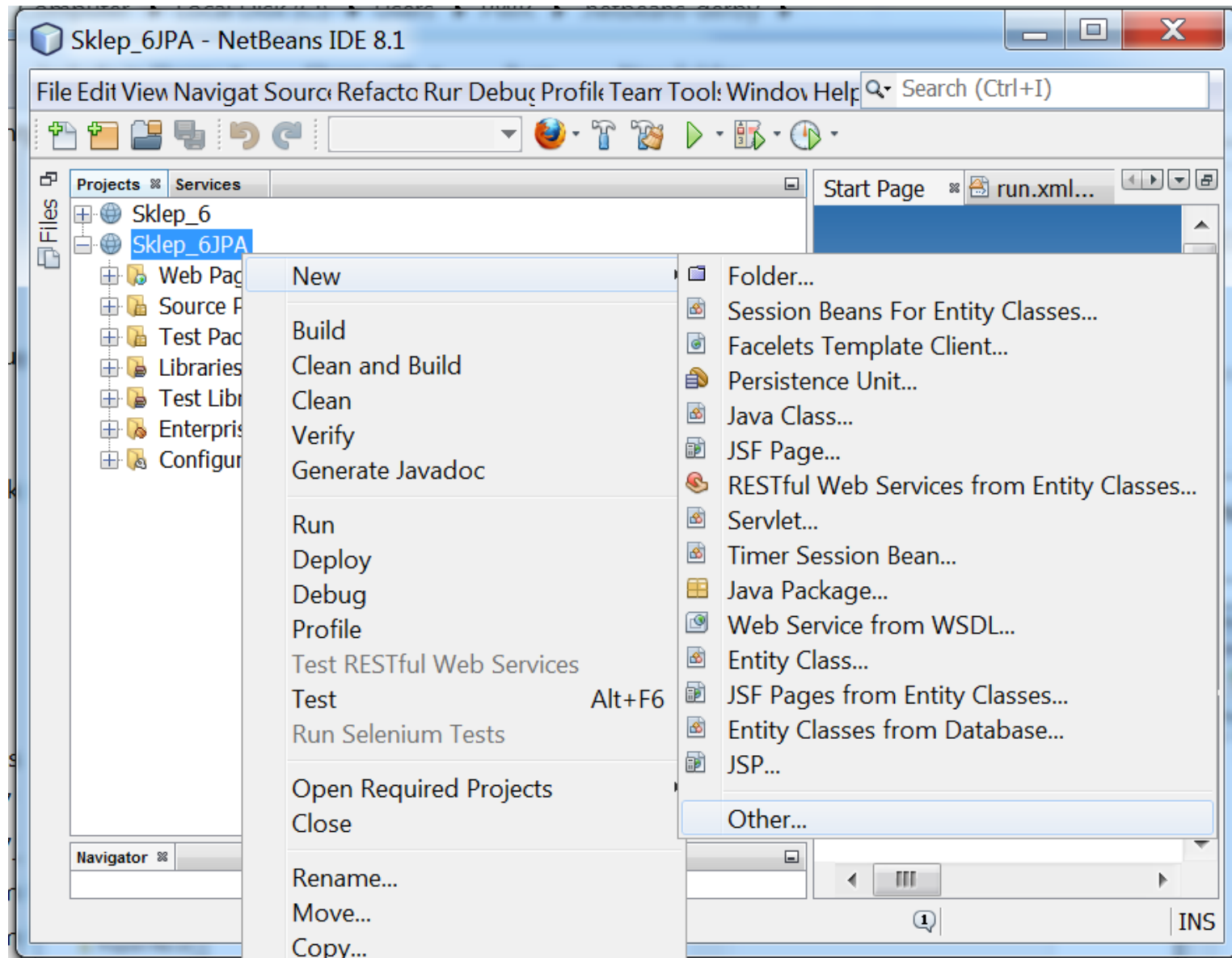
## 4. Połączenie z pustą bazą danych (wybór pozycji Connect po kliknięciu prawym klawiszem myszy na połączeniu)



## 5. Widok utworzonej pustej bazy danych **Produkt**

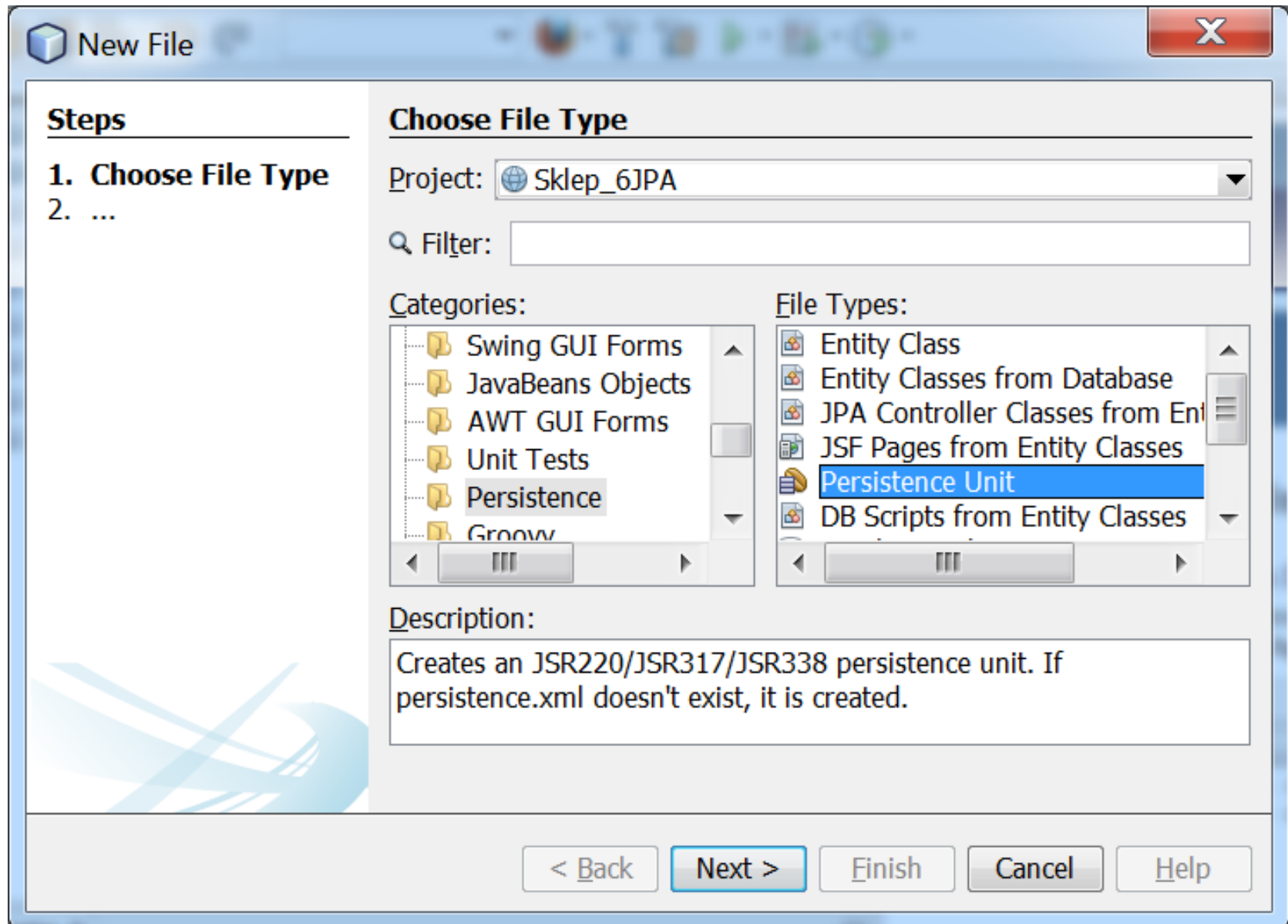


## 6. Dodanie pliku **persistence.xml** do projektu, potrzebnego w procesie utrwalania danych metodą ORM - **New/Other**

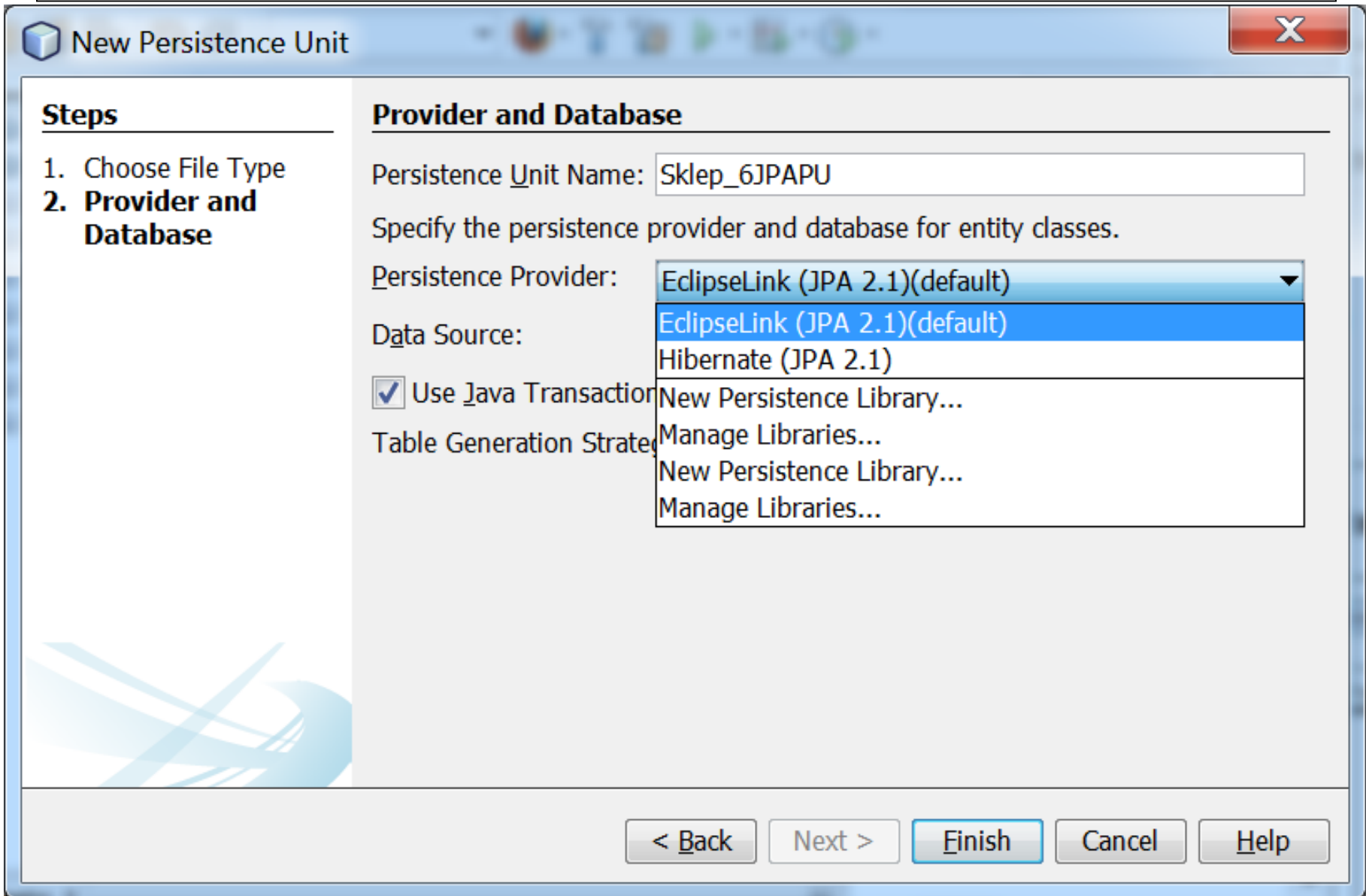


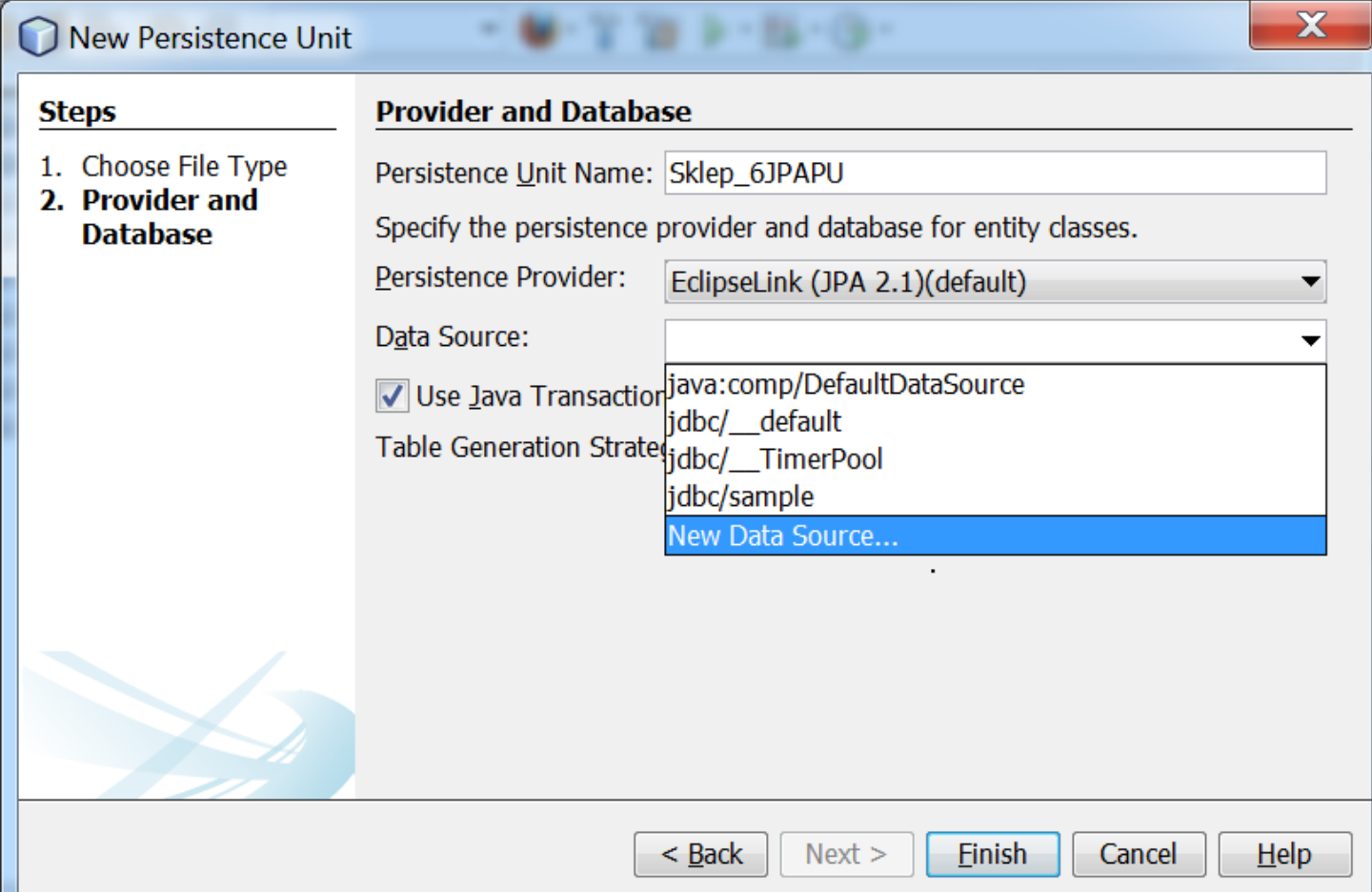


## 7. Wybór Persistence/Persistence Unit



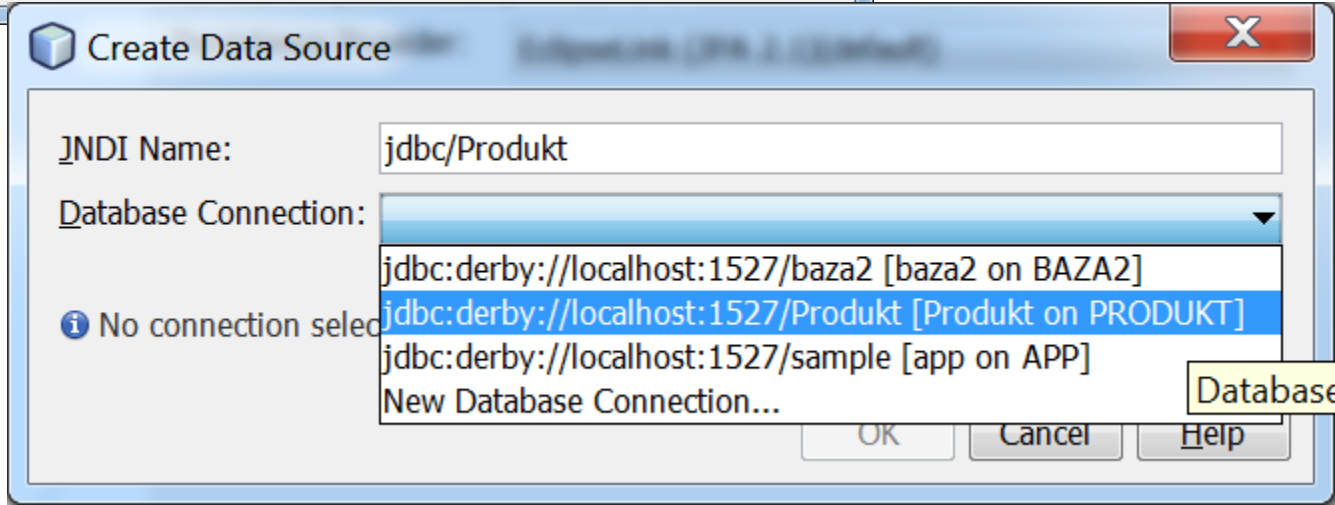
## 8. Wybór technologii EclipseLink do obsługi bazy danych metodą ORM





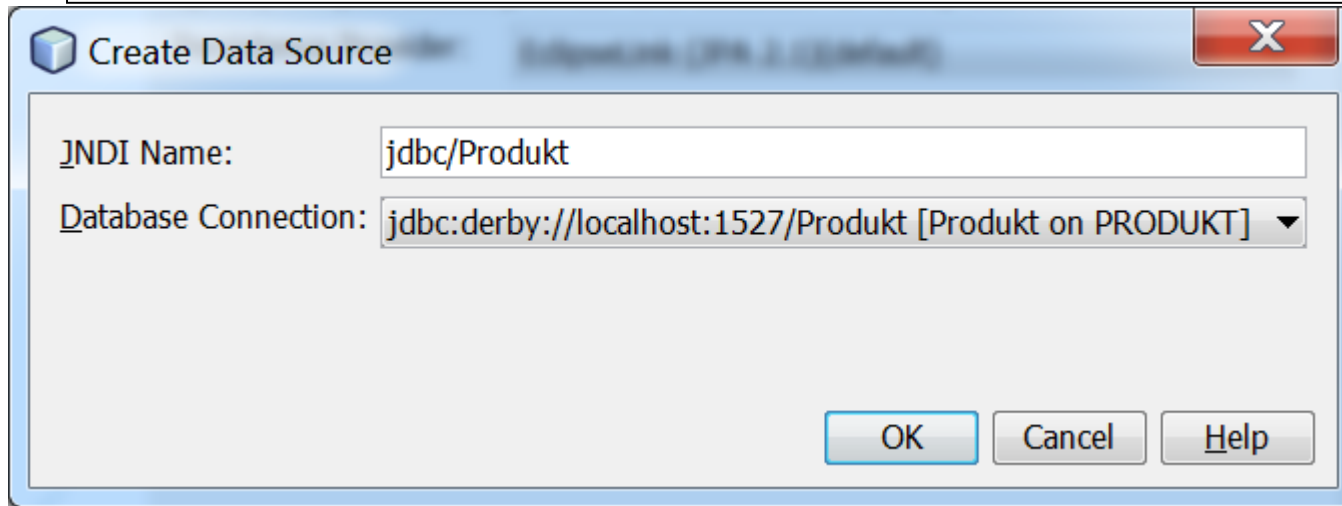
9. Utworzenie połączenia opartego na JNDI – nadanie nazwy **jdbc/Produkt**

Wybór połączenia z bazą danych



Database

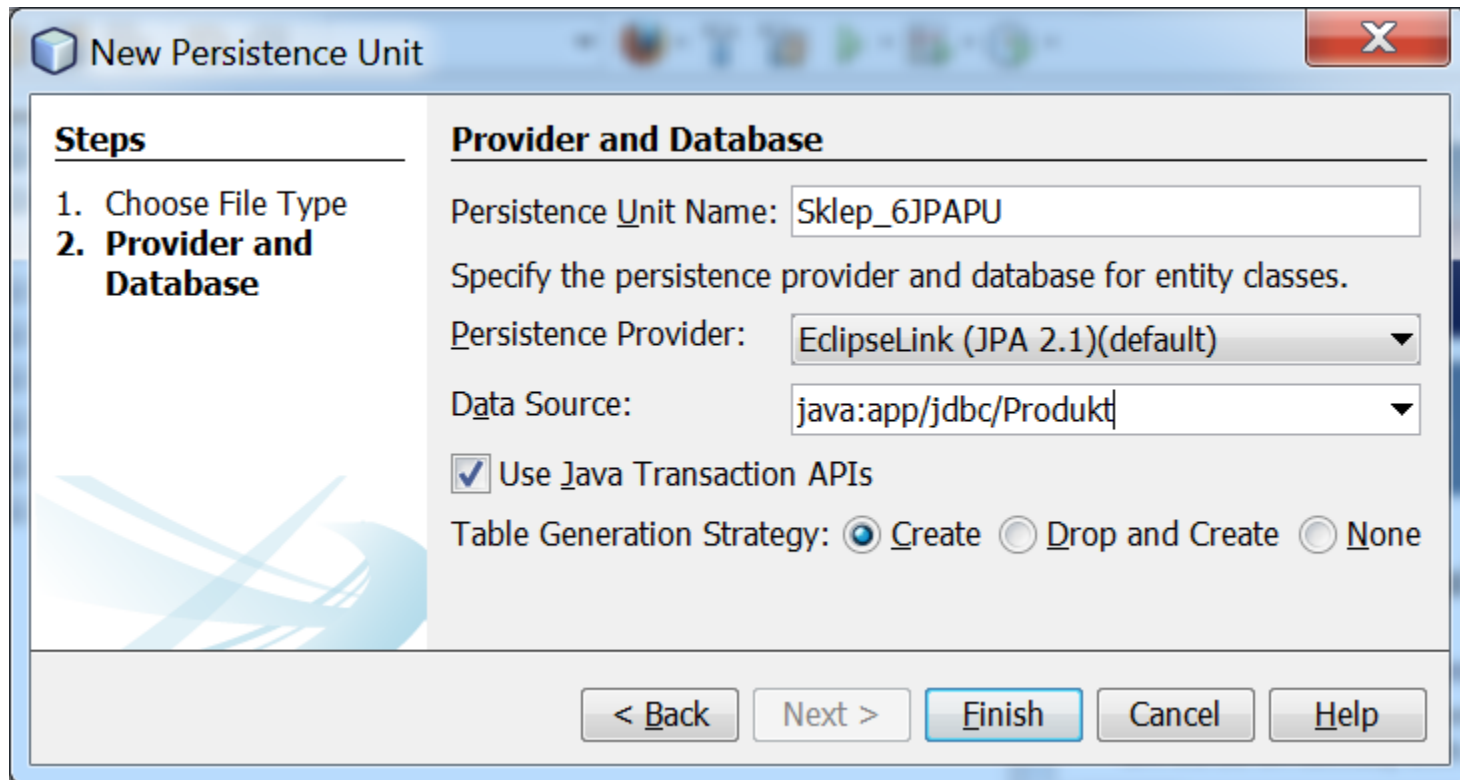
## 10. Gotowe połączenie z bazą danych - Finish



The 'Create Data Source' dialog box is shown with the following fields:

- JNDI Name:** jdbc/Produkt
- Database Connection:** jdbc:derby://localhost:1527/Produkt [Produkt on PRODUKT]

Buttons: OK, Cancel, Help

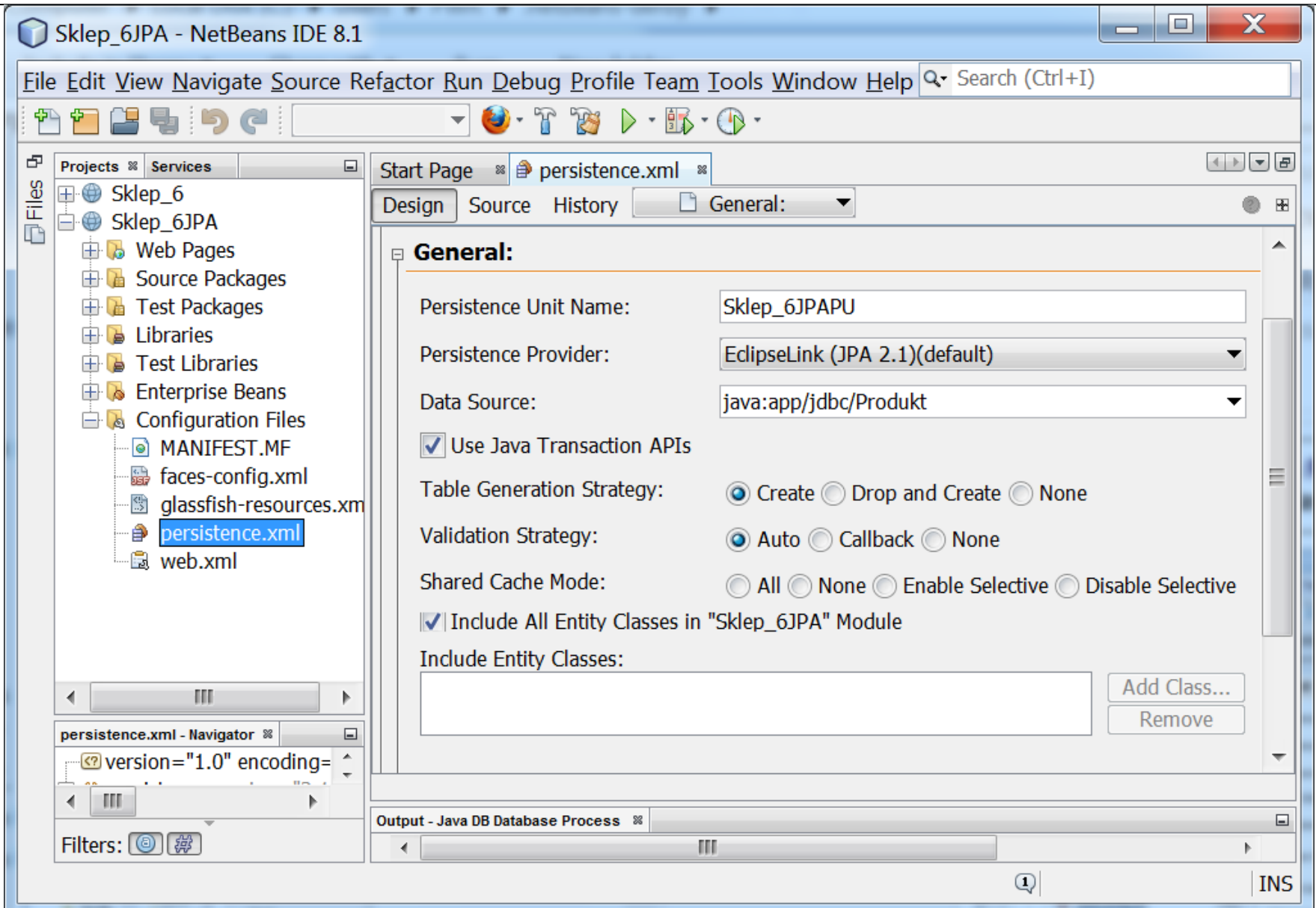


The 'New Persistence Unit' dialog box is shown with the following fields and options:

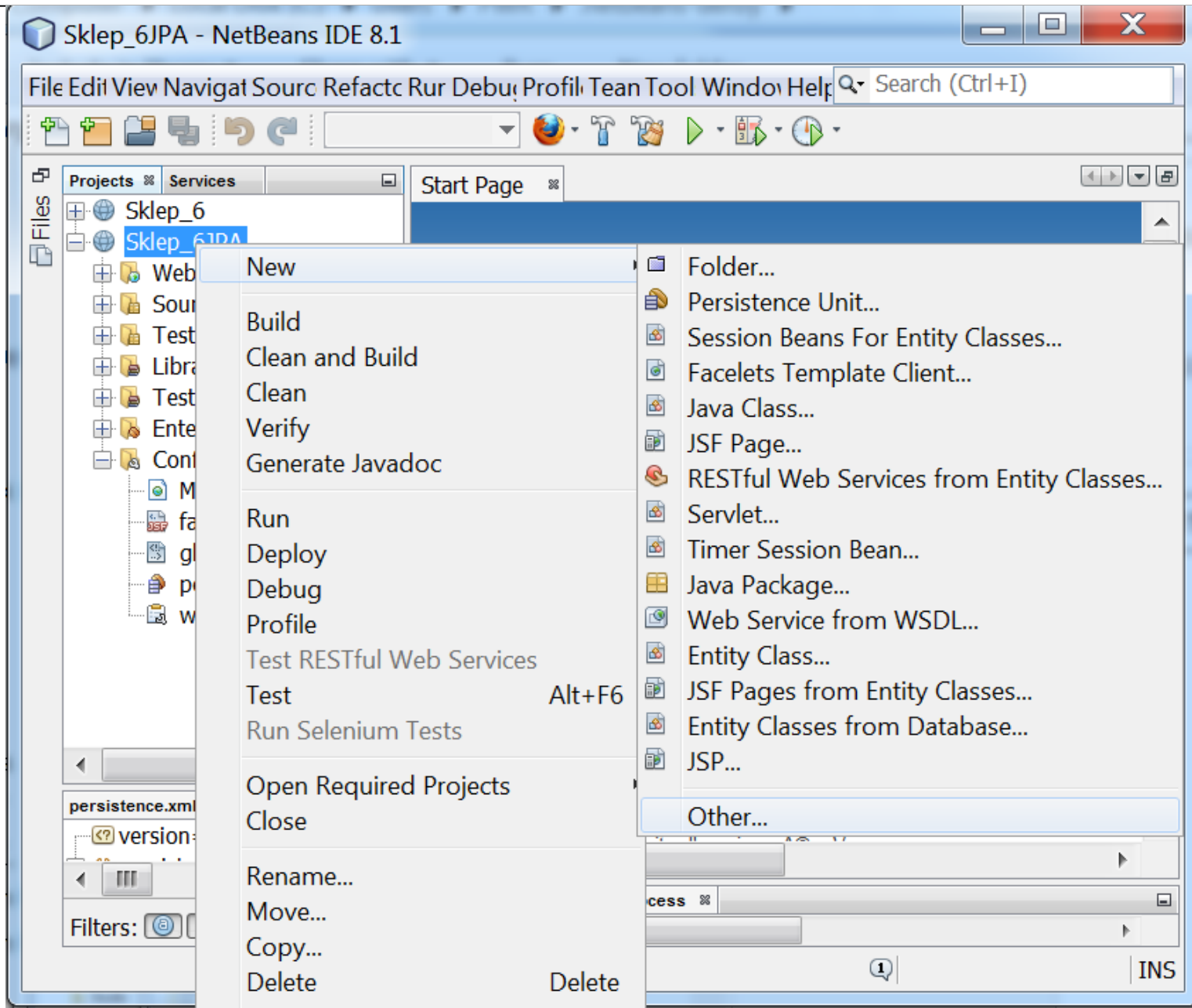
- Steps:**
  1. Choose File Type
  2. **Provider and Database**
- Provider and Database:**
  - Persistence Unit Name:** Sklep\_6JPAPU
  - Persistence Provider:** EclipseLink (JPA 2.1)(default)
  - Data Source:** java:app/jdbc/Produkt
  - Use Java Transaction APIs
  - Table Generation Strategy:**  Create  Drop and Create  None

Buttons: < Back, Next >, Finish, Cancel, Help

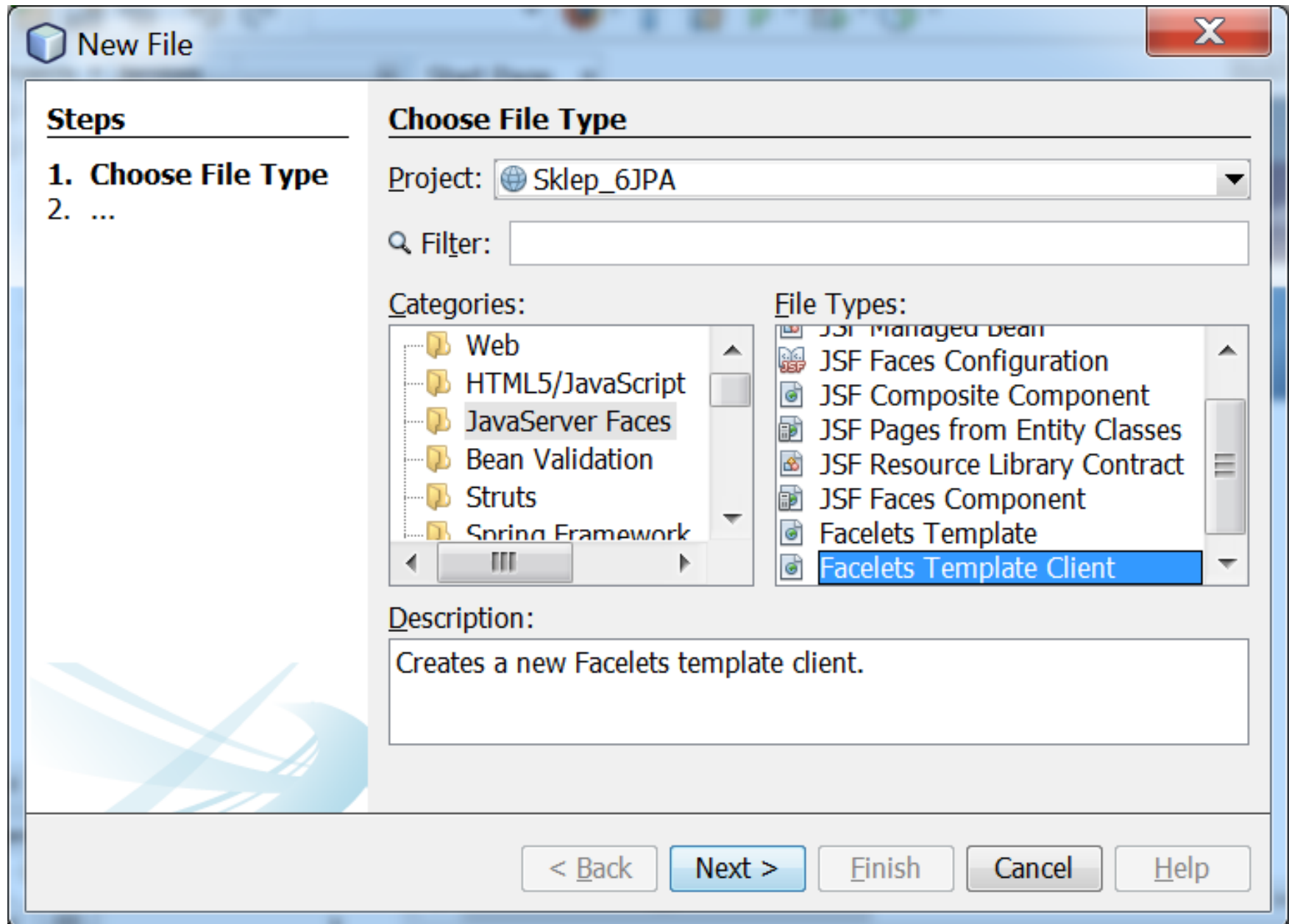
# 11. Pełna zawartość pliku **Persistence.xml**, pozwalająca na korzystanie z technologii ORM dla klas z adnotacją **@Entity** (tutaj klasa **Produkt**)



## 12. Dodanie nowej strony do obsługi bazy danych – **New/Other**



## 13. Dodanie strony opartej na szablonie **template.xml** (1)



## 14. Dodanie strony **baza.xhtml** opartej na szablonie **template.xml** (2)

New Facelets Template Client

**Steps**

1. Choose File Type
2. **Name and Location**

**Name and Location**

File Name:

Project:

Folder:

Created File:

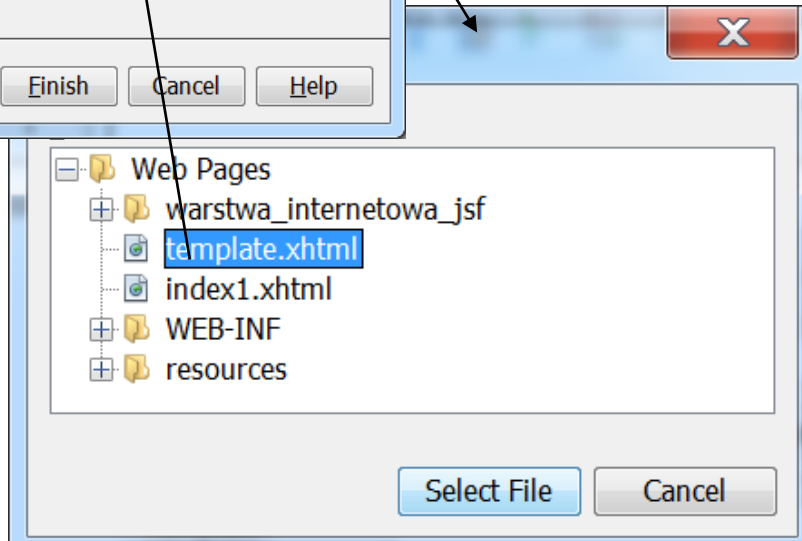
Template:

Generated Root Tag:  <html>  
 <ui:composition>

Sections To Generate:

<input checked="" type="checkbox"/>	title
<input checked="" type="checkbox"/>	top
<input checked="" type="checkbox"/>	content
<input checked="" type="checkbox"/>	bottom

< Back   Next >   Finish   Cancel   Help





## 15. Zawartość wygenerowanego pliku **baza.xhtml**

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
    <html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
  xmlns:h="http://xmlns.jcp.org/jsf/html">
<body>
  <ui:composition template=" ../template.xhtml">
    <ui:define name="title">
      <h:outputText value="#{bundle['jsf.baza.tytul']}"></h:outputText>
    </ui:define>
```

## 16. Modyfikacja zawartości strony **baza.xhtml** opartej na szablonie **template.xml**

```
<ui:define name="content">
  <h:form>
    <h:commandLink action="#{managed_produkt.zapisz}"
      value="#{bundle['jsf.baza.zapisz']}" /><br/>
    <h:commandLink action="#{managed_produkt.pobierz}"
      value="#{bundle['jsf.baza.pobierz']}" /><br/>
  </h:form>
</ui:define>
</ui:composition>
</body>
</html>
```

Pierwszy comandLink obsługuje **zapis** do bazy danych, a drugi **odczyt** z bazy danych

## 17. Uzupełnienie pliku **Bundle.properties**

jsf.baza.tytul=Utrwalanie danych

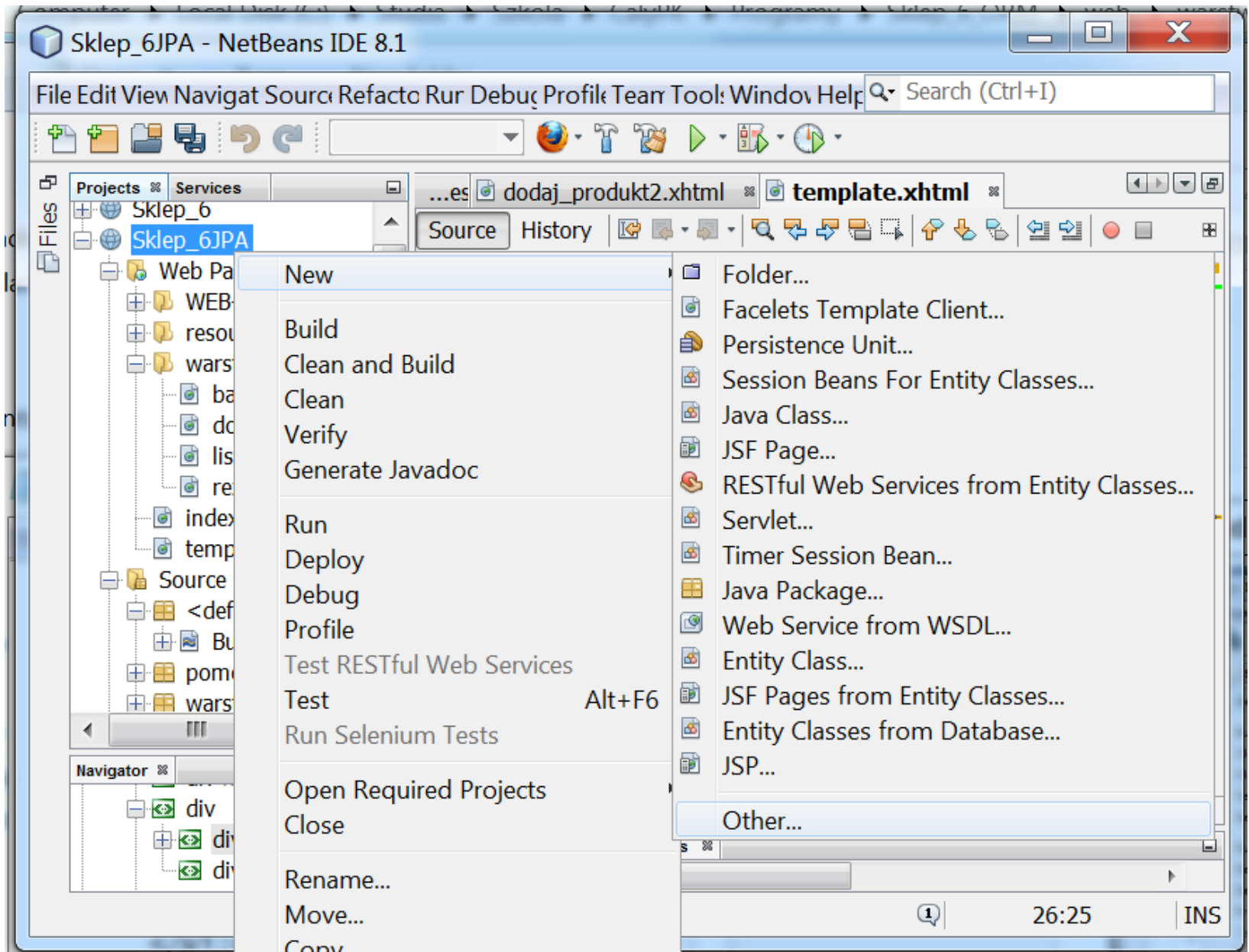
jsf.baza.zapisz=Zapisz do bazy danych

jsf.baza.pobierz=Pobierz z bazy danych

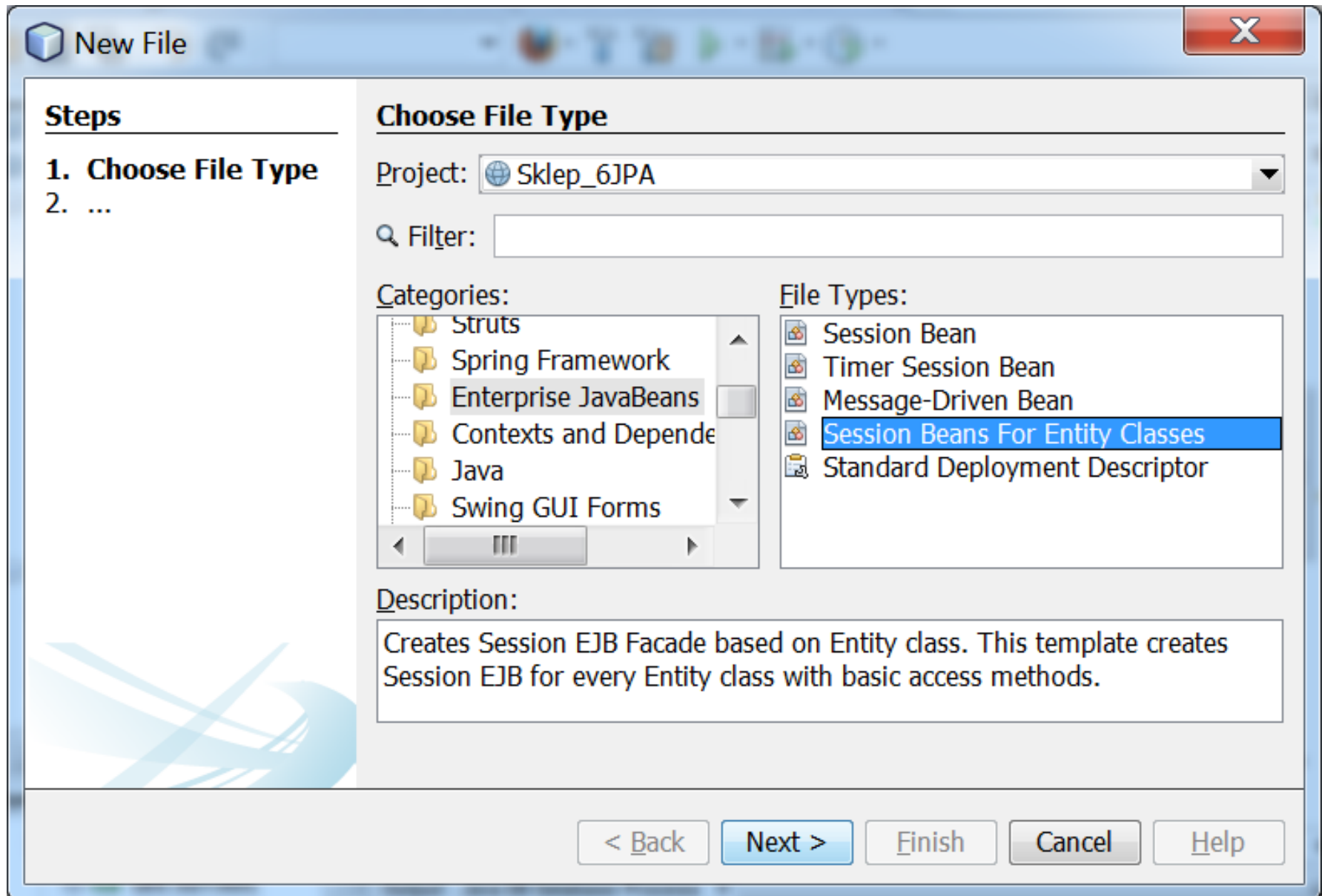
18. Uzupełnienie kodu pliku **template.xhtml** – bloku "left", czyli dodanie znacznika **h:link** do wywoływania strony **baza.xhtml**

```
<div id="left">  
    <h:link outcome="/faces/jsf/dodaj_produkt2"  
        value="Dodaj produkt"/> <br/>  
    <h:link outcome="/faces/jsf/lista_produktow"  
        value="Lista produktow"/><br/>  
    <h:link outcome="/faces/warstwa_internetowa_jsf/baza"  
        value="#{bundle['jsf.baza.tytul']}" />  
</div>
```

## 19. Dodanie klasy do utrwalania obiektów typu Produkt – **New/Other**



## 20. Utworzenie pliku wspierającego ORM klasy Produkt – **Enterprise JavaBeans/SessionBeans for Entity Classes**



## 21. Wskazanie klasy Produkt1 typu Entity

The image shows two screenshots of the 'New Session Beans For Entity Classes' dialog box. The top screenshot shows the initial state where the 'Available Entity Classes' list contains 'warstwa\_biznesowa.entity' and the 'Selected Entity Classes' list is empty. The bottom screenshot shows the state after selecting the 'Produkt1' class, which is now listed in the 'Selected Entity Classes' list.

**New Session Beans For Entity Classes**

**Steps**

1. Choose File Type
- 2. Entity Classes**
3. Generated Session Beans

**Entity Classes**

Available Entity Classes:

warstwa\_biznesowa.entity

Selected Entity Classes:

Include Referenced Classes

**Buttons:** < Back, Next >, Finish, Cancel, Help

---

**New Session Beans For Entity Classes**

**Steps**

1. Choose File Type
- 2. Entity Classes**
3. Generated Session Beans

**Entity Classes**

Available Entity Classes:

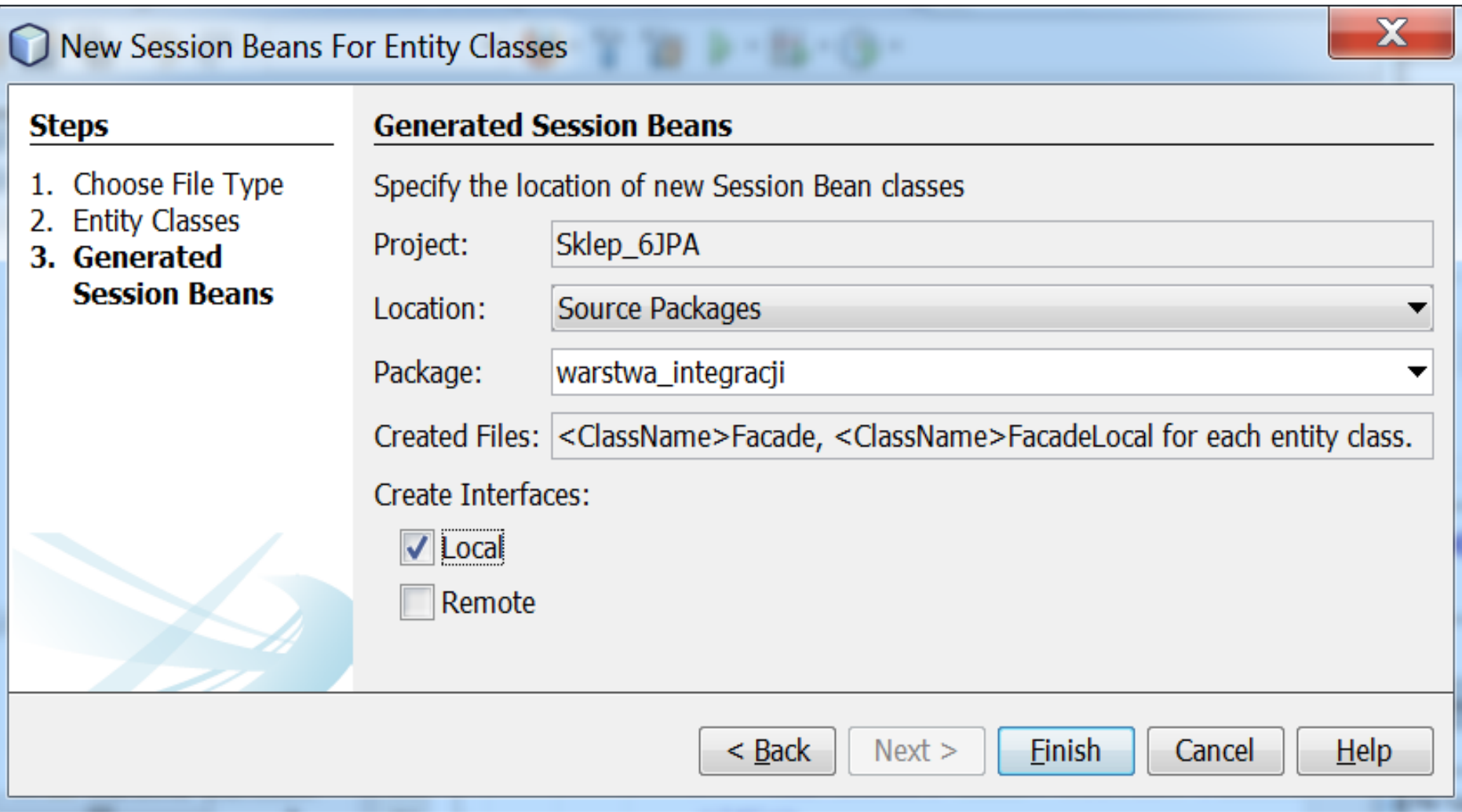
Selected Entity Classes:

warstwa\_biznesowa.entity.Produkt1

Include Referenced Classes

**Buttons:** < Back, Next >, Finish, Cancel, Help

## 22. Wybór **Local** jako zasięg dla klasy typu EJB oraz wpisanie nazwy orm w polu Package



**New Session Beans For Entity Classes**

**Steps**

1. Choose File Type
2. Entity Classes
- 3. Generated Session Beans**

**Generated Session Beans**

Specify the location of new Session Bean classes

Project: Sklep\_6JPA

Location: Source Packages

Package: warstwa\_integracji

Created Files: <ClassName>Facade, <ClassName>FacadeLocal for each entity class.

Create Interfaces:

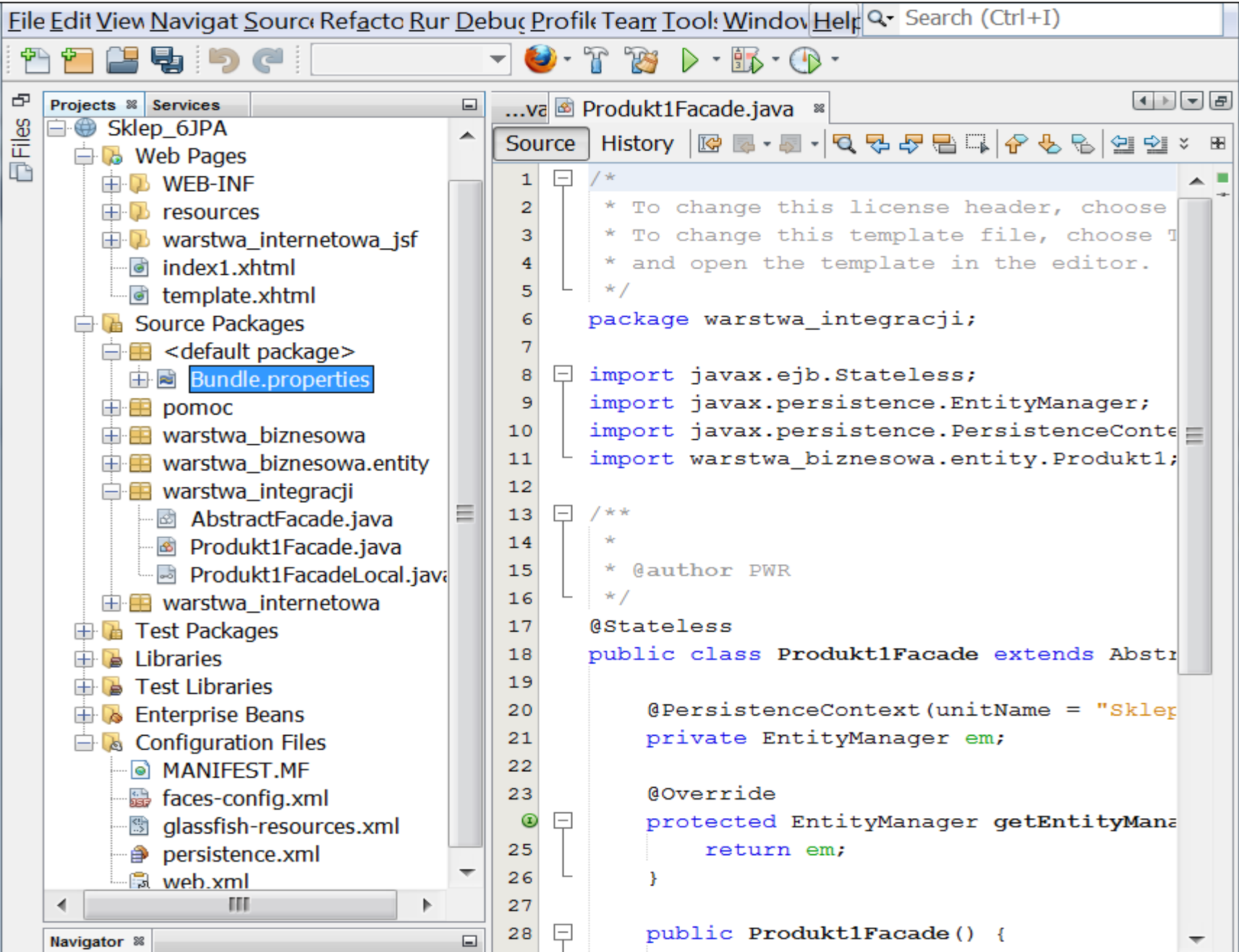
Local

Remote

< Back   Next >   **Finish**   Cancel   Help



## 23. Widok wygenerowanego interfejsu **Produkt1FacadeLocal**



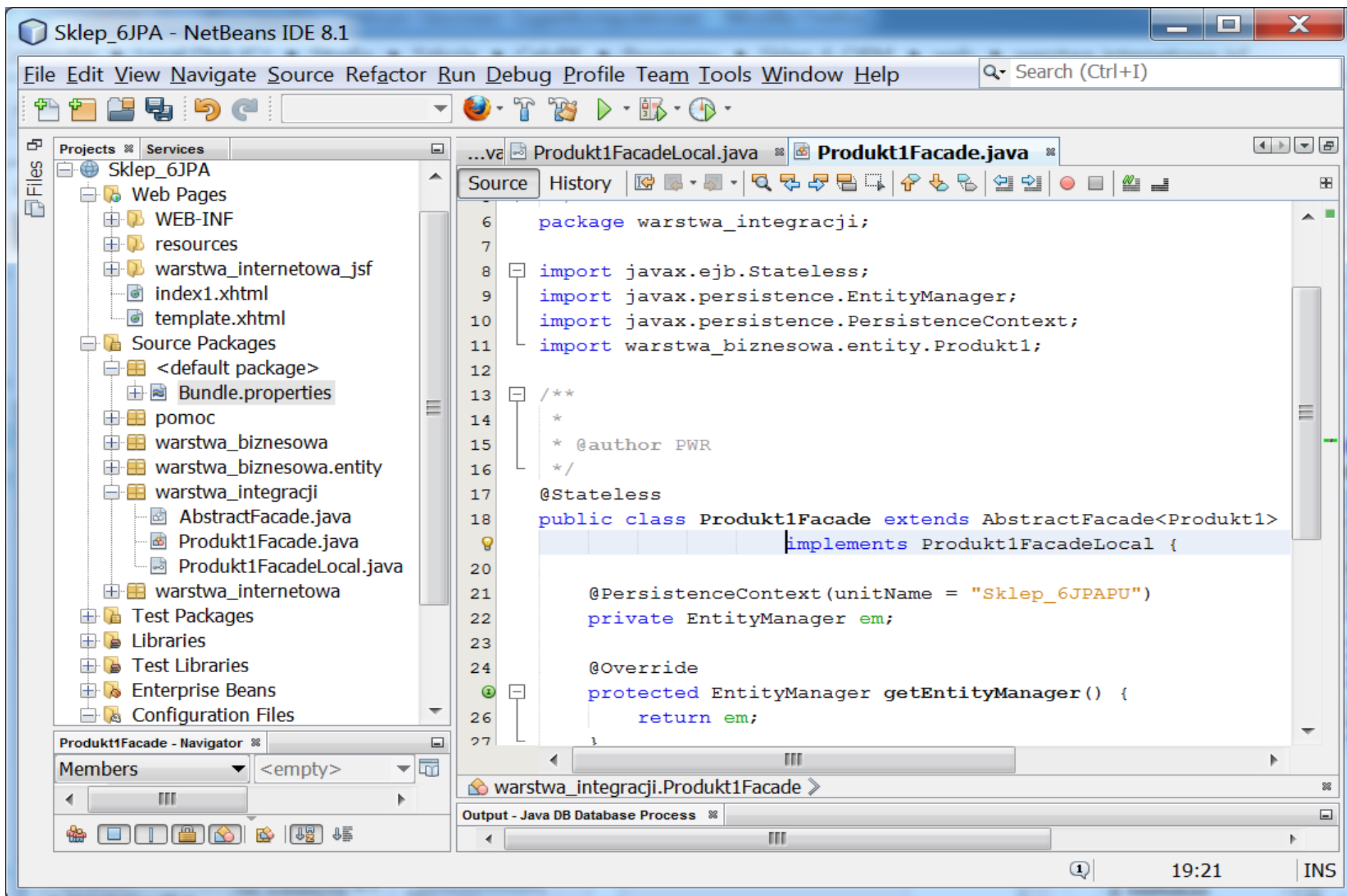
The screenshot displays an IDE window with the following components:

- Menu Bar:** File, Edit, View, Navigat, Sourc, Refacto, Rur, Debuç, Profil, Tear, Tool, Window, Help. A search bar with "Search (Ctrl+I)" is also present.
- Toolbar:** Contains icons for file operations (new, open, save, print), navigation (back, forward), and execution (run, debug).
- Left Pane (Project Explorer):** Shows a project named "Sklep\_6JPA". The tree structure includes:
  - Web Pages
    - WEB-INF
    - resources
    - warstwa\_internetowa\_jsf
      - index1.xhtml
      - template.xhtml
  - Source Packages
    - <default package>
      - Bundle.properties (highlighted)
      - pomoc
      - warstwa\_biznesowa
        - warstwa\_biznesowa.entity
        - warstwa\_integracji
          - AbstractFacade.java
          - Produkt1Facade.java
          - Produkt1FacadeLocal.java
      - warstwa\_internetowa
    - Test Packages
    - Libraries
    - Test Libraries
    - Enterprise Beans
    - Configuration Files
      - MANIFEST.MF
      - faces-config.xml
      - glassfish-resources.xml
      - persistence.xml
      - web.xml

- Right Pane (Code Editor):** Shows the source code for "Produkt1Facade.java". The code is as follows:

```
1  /*
2  * To change this license header, choose
3  * To change this template file, choose T
4  * and open the template in the editor.
5  */
6  package warstwa_integracji;
7
8  import javax.ejb.Stateless;
9  import javax.persistence.EntityManager;
10 import javax.persistence.PersistenceConte
11 import warstwa_biznesowa.entity.Produkt1;
12
13 /**
14 *
15 * @author PWR
16 */
17 @Stateless
18 public class Produkt1Facade extends Abstr
19
20     @PersistenceContext(unitName = "Sklep
21     private EntityManager em;
22
23     @Override
24     protected EntityManager getEntityMana
25         return em;
26     }
27
28     public Produkt1Facade() {
```

## 24. Widok kontrolera ORM **Produkt1Facade**, który implementuje interfejs **Produkt1FacadeLocal** i dziedziczy po klasie abstrakcyjnej **AbstractFacade**



The screenshot displays the NetBeans IDE 8.1 interface. The main editor window shows the source code for `Produkt1Facade.java` in the `warstwa_integracji` package. The code implements the `Produkt1FacadeLocal` interface and inherits from the `AbstractFacade` class.

```
6 package warstwa_integracji;
7
8 import javax.ejb.Stateless;
9 import javax.persistence.EntityManager;
10 import javax.persistence.PersistenceContext;
11 import warstwa_biznesowa.entity.Produkt1;
12
13 /**
14  *
15  * @author PWR
16  */
17 @Stateless
18 public class Produkt1Facade extends AbstractFacade<Produkt1>
19     implements Produkt1FacadeLocal {
20
21     @PersistenceContext(unitName = "Sklep_6JPAPU")
22     private EntityManager em;
23
24     @Override
25     protected EntityManager getEntityManager() {
26         return em;
27     }
28 }
```

The interface `Produkt1FacadeLocal` is visible in the background, showing the `getEntityManager()` method signature.

The IDE interface includes a Project Explorer on the left showing the project structure for `Sklep_6JPA`, with the `warstwa_integracji` package selected. The bottom status bar shows the time as 19:21 and the cursor position as INS.

## 25. Wygenerowana klasa abstrakcyjna **AbstractFacade** (1)

The screenshot displays the NetBeans IDE 8.1 interface. The main window shows the source code of the `AbstractFacade` class in the `warstwa_integracji` package. The code is as follows:

```
1  ...5 lines
6  package warstwa_integracji;
7  import java.util.List;
8  import javax.persistence.EntityManager;
9
10 /**...4 lines */
11 public abstract class AbstractFacade<T> {
12     private Class<T> entityClass;
13     public AbstractFacade(Class<T> entityClass) {
14         this.entityClass = entityClass;
15     }
16     protected abstract EntityManager getEntityManager();
17
18     public void create(T entity) {
19         getEntityManager().persist(entity);
20     }
21     public void edit(T entity) {
22         getEntityManager().merge(entity);
23     }
24     public void remove(T entity) {
25         getEntityManager().remove(getEntityManager().merge(entity));
26     }
27 }
28
29
```

The IDE interface includes a menu bar (File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help), a search bar (Search (Ctrl+I)), a toolbar with various icons, and a project explorer on the left showing the project structure for `Sklep_6JPA`. The `AbstractFacade.java` file is selected in the project explorer. The bottom status bar shows the current cursor position at line 27, column 1, in Insert mode (INS).

## 26. Wygenerowana klasa abstrakcyjna **AbstractFacade** (2)

```
...es dodaj_produk2.xhtml  template.xhtml  AbstractFacade.java  Produkt1FacadeLocal.java  Produkt1Facade.java
Source History
30 public T find(Object id) {
31     return getEntityManager().find(entityClass, id);
32 }
33 public List<T> findAll() {
34     javax.persistence.criteria.CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery();
35     cq.select(cq.from(entityClass));
36     return getEntityManager().createQuery(cq).getResultList();
37 }
38 public List<T> findRange(int[] range) {
39     javax.persistence.criteria.CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery();
40     cq.select(cq.from(entityClass));
41     javax.persistence.Query q = getEntityManager().createQuery(cq);
42     q.setMaxResults(range[1] - range[0] + 1);
43     q.setFirstResult(range[0]);
44     return q.getResultList();
45 }
46 public int count() {
47     javax.persistence.criteria.CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery();
48     javax.persistence.criteria.Root<T> rt = cq.from(entityClass);
49     cq.select(getEntityManager().getCriteriaBuilder().count(rt));
50     javax.persistence.Query q = getEntityManager().createQuery(cq);
51     return ((Long) q.getSingleResult()).intValue();
52 }
53 }
```

## 27. Utworzenie referencji do ziarna **Produkt1Facade** w klasie **Fasada\_warstwy\_biznesowej** (1)

The screenshot shows the NetBeans IDE 8.1 interface. The main editor displays the source code of `Fasada_warstwy_biznesowej.java` in the `warstwa_biznesowa` package. The code includes imports for `java.util.ArrayList`, `java.util.Date`, `javax.ejb.Stateless`, and `warstwa_biznesowa.entity.Produkt1`. The class is annotated with `@Stateless` and contains a `produkty` field of type `ArrayList<Produkt1>` and a `stan` field of type `boolean`. A context menu is open over the code, listing various actions and their keyboard shortcuts:

Action	Keyboard Shortcut
Navigate	
Show Javadoc	Alt+F1
Find Usages	Alt+F7
Call Hierarchy	
Insert Code...	Alt+Insert
Fix Imports	Ctrl+Shift+I
Refactor	
Format	Alt+Shift+F
Run File	Shift+F6
Debug File	Ctrl+Shift+F5
Test File	Ctrl+F6
Debug Test File	Ctrl+Shift+F6
Run Focused Test Method	
Debug Focused Test Method	
Run Into Method	
New Watch...	Ctrl+Shift+F7
Toggle Line Breakpoint	Ctrl+F8
Profile	

28. Utworzenie referencji do ziarna **Produkt1Facade** w klasie **Fasada\_warstwy\_biznesowej** (2) – **Insert Code/Call Enterprise Bean** i wybór z listy ziarna typu **Produkt1Facade** – pojawiła się referencja typu **Produkt1FacadeLocal**

The screenshot displays the NetBeans IDE 8.1 interface. The main editor window shows the source code of the class `Fasada_warstwy_biznesowej.java`. The code includes imports for `javax.ejb.Stateless`, `warstwa_biznesowa.entity.Produkt1`, and `warstwa_integracji.Produkt1FacadeLocal`. The class is annotated with `@Stateless` and `@EJB`. A private field `produkt1Facade` of type `Produkt1FacadeLocal` is declared on line 14. The code also includes a static variable `klucz`, a private list `produkty`, a boolean `stan`, and two public methods: `getProdukty()` and `setProdukty()`.

The **Call Enterprise Bean** dialog is open in the foreground. It prompts the user to "Select an enterprise bean from open projects." The project tree shows the following structure:

- Sklep\_6
- Sklep\_6JPA
  - Fasada\_warstwy\_biznesowej
    - Produkt1Facade** (selected)

29. Przekazanie nadawania **id** klasie Produkt1 typu Entity mechanizmowi utrwalania JPA – modyfikacja kodu dodawania obiektu typu Produkt1 (Entity) w komponencie EJB Fasada\_warstwy\_biznesowej

```
public void utworz_produkt(String dane[], Date data) {  
    Produkt1 produkt = new Produkt1();  
    // klucz++;  
    // produkt.setId(new Long(klucz));  
    produkt.setNazwa(dane[0]);  
    produkt.setCena(Float.parseFloat(dane[1]));  
    produkt.setPromocja(Integer.parseInt(dane[2]));  
    produkt.setData_produkcji(data);  
    dodaj_produkt(produkt);  
}
```

gdzie w klasie typu Produkt1 atrybut: **strategy = GenerationType.AUTO** oznacza możliwość generowania klucza głównego podczas utrwalania przez aplikację lub przez komponenty JPA

**@Id**

**@GeneratedValue(strategy = GenerationType.AUTO)**

private Long id;

29 (cd). Przekazanie nadawania id klasie Produkt1 typu Entity mechanizmom utrwalania JPA – modyfikacja kodu dodawania obiektu typu Produkt1 (Entity) w komponencie EJB Fasada\_warstwy\_biznesowej

```
public ArrayList<ArrayList<String>> items() {
    ArrayList<ArrayList<String>> dane = new ArrayList();
    for (Produkt1 p : produkty) {
        ArrayList<String> wiersz = new ArrayList();
        wiersz.add(p.getId_().toString());
        wiersz.add(p.getNazwa());
        wiersz.add("" + p.getCena());
        wiersz.add("" + p.getPromocja());
        wiersz.add(p.getData_produkcji().toString());
        wiersz.add("" + p.cena_brutto());
        dane.add(wiersz);
    }
    return dane;
}
```

gdzie w obiekcie typu Produkt1, przed zapisaniem do bazy danych **id** jest równe null:

```
public Long getId_() {
    if(id==null)
        return new Long(0);
    return id;
}
```



30. Metody w klasie **Fasada\_wartwy\_biznesowej** do utrwalania danych i do pobierania danych z danych Produkt1.

```
public void zapisz() {  
    for (Produkt1 p : produkty) {  
        Long id = p.getId();  
        if (id == null || produkt1Facade.find(p.getId()) == null)  
            produkt1Facade.create(p);  
    }  
}  
  
public void pobierz() {  
    List<Produkt1> pom = produkt1Facade.findAll();  
    produkty.clear();  
    produkty.addAll(pom);  
}
```

## 31. Metody w klasie **Managed\_produk**t

```
public String zapisz() {  
    fasada.zapisz();  
    return "/faces/index1";  
}
```

```
public String pobierz() {  
    fasada.pobierz();  
    return "/faces/index1";  
}
```

Aktualizacja liczby stron po odczytaniu danych z bazy danych

Aktualizacja modelu tabeli

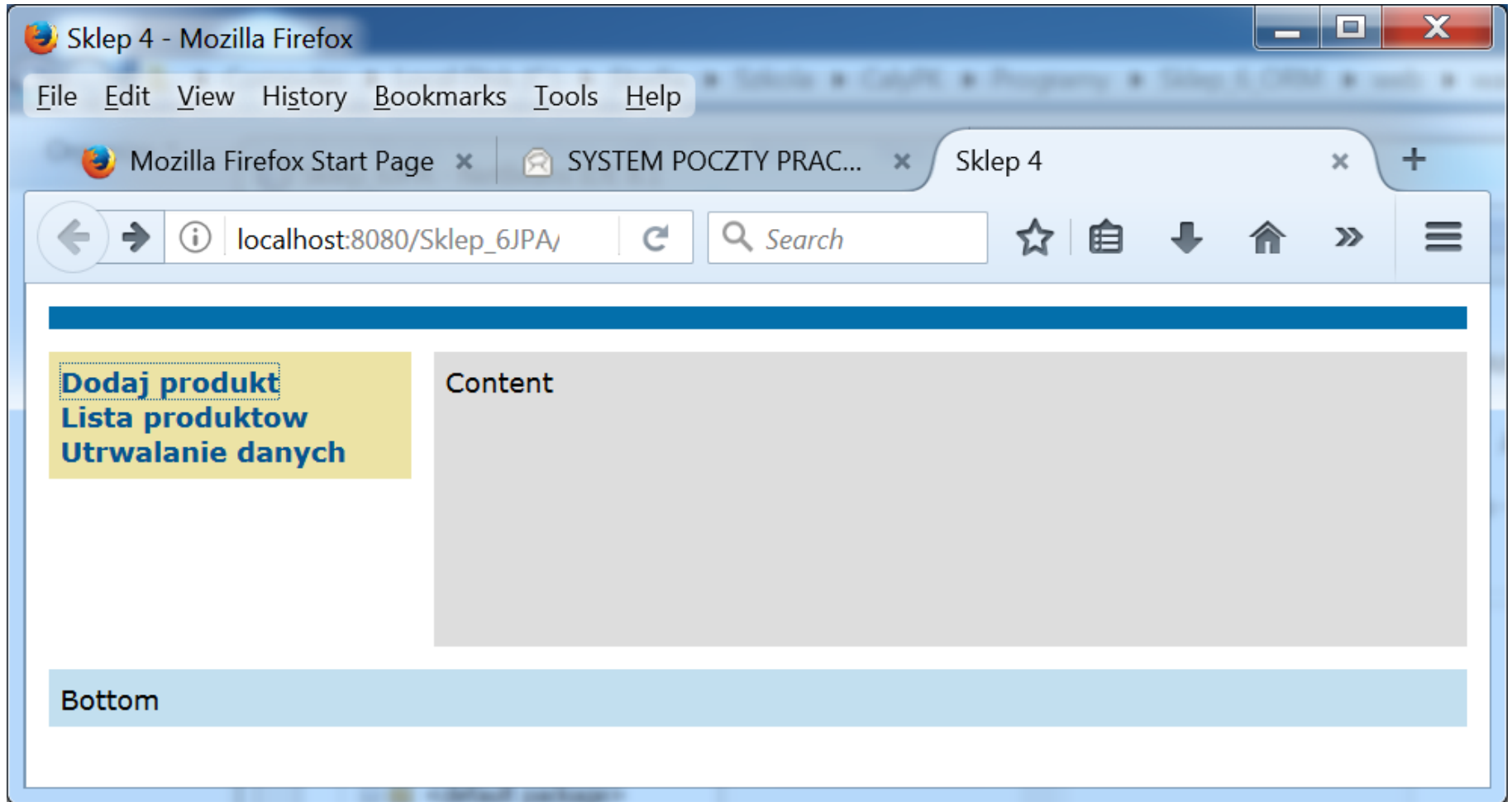
## 32. Baza danych jest pusta przed uruchomieniem aplikacji

The screenshot displays the NetBeans IDE 8.1 interface for a project named "Sklep\_6JPA".

- Projects Window:** Shows a tree view of the project structure. Under "Databases", there is a connection "jdbc:derby://localhost:1527/Produkt [Produkt]". Below it, a folder named "PRODUKT" is expanded, showing "Tables", "Views", and "Procedures".
- Source Editor:** The file "Managed\_produkty.java" is open. The code shows a facade class with methods "zapisz()" and "pobierz()".

```
164 ((UIInput) toValidate).s
165 FacesMessage message = n
166 context.addMessage(toVal:
167 stan = 0;
168 }
169 }
170
171 public String zapisz() {
172     fasada.zapisz();
173     return "/faces/index1";
174 }
175
176 public String pobierz() {
177     fasada.pobierz();
178     return "/faces/index1";
179 }
180
181 }
182
```
- Managed\_produkty - Navigator:** Shows the class "Managed\_produkty :: Serializable, ActionListener" with a member "Managed\_produkty()".
- Output Window:** Shows the output of the "Java DB Database Process" and "Sklep\_6JPA (clean,dist)".
- Status Bar:** Shows "180:1 | INS".

## 33. Uruchomienie aplikacji



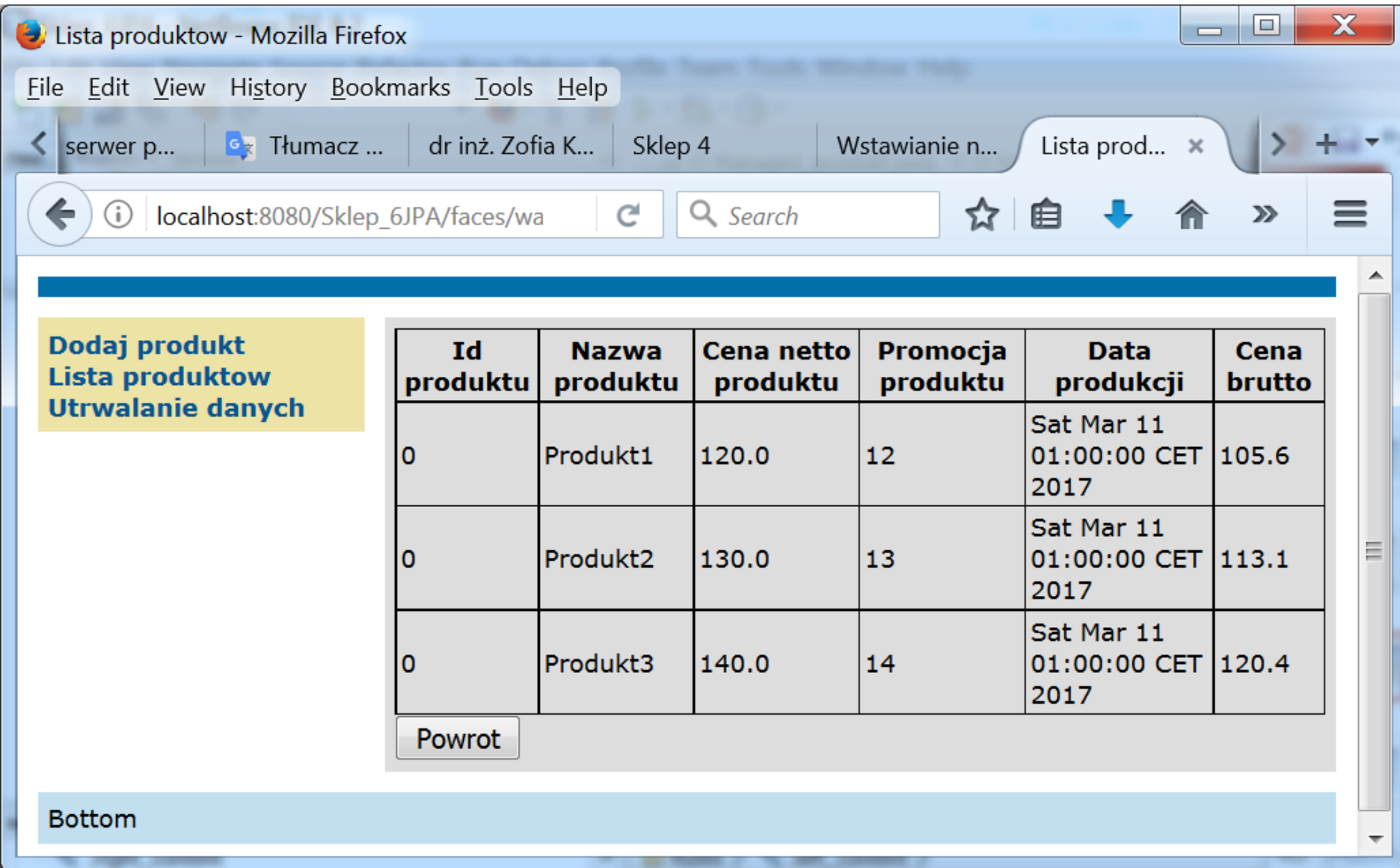
## 34. Po uruchomieniu aplikacji tworzy się schemat bazy danych.

The screenshot displays the NetBeans IDE 8.1 interface. On the left, the 'Projects' pane shows a tree view of a database schema for 'PRODUKT'. The schema includes tables: PRODUKT1 (with columns ID, CENA, DATA\_PRODUKCJI, NAZWA, PROMOCJA), Indexes, Foreign Keys, SEQUENCE, Views, and Procedures. Below the database view, the 'Navigator' pane shows 'CSS' and 'HTML' files. The main editor window on the right shows an HTML file named 'template.xhtml...' with the following code:

```
16 <div id="to
17 <h:pane
18 <ui
19 </h:pane
20 </div>
21 <div>
22 <div id:
23 <h:
24 <h:
25 <h:
26
27
28 </div>
29 <div id:
30 <ui
31 </div>
32 </div>
```

At the bottom of the IDE, the 'Output' window shows 'Java DB Database Process' and the status bar indicates '25:64' and 'INS'.

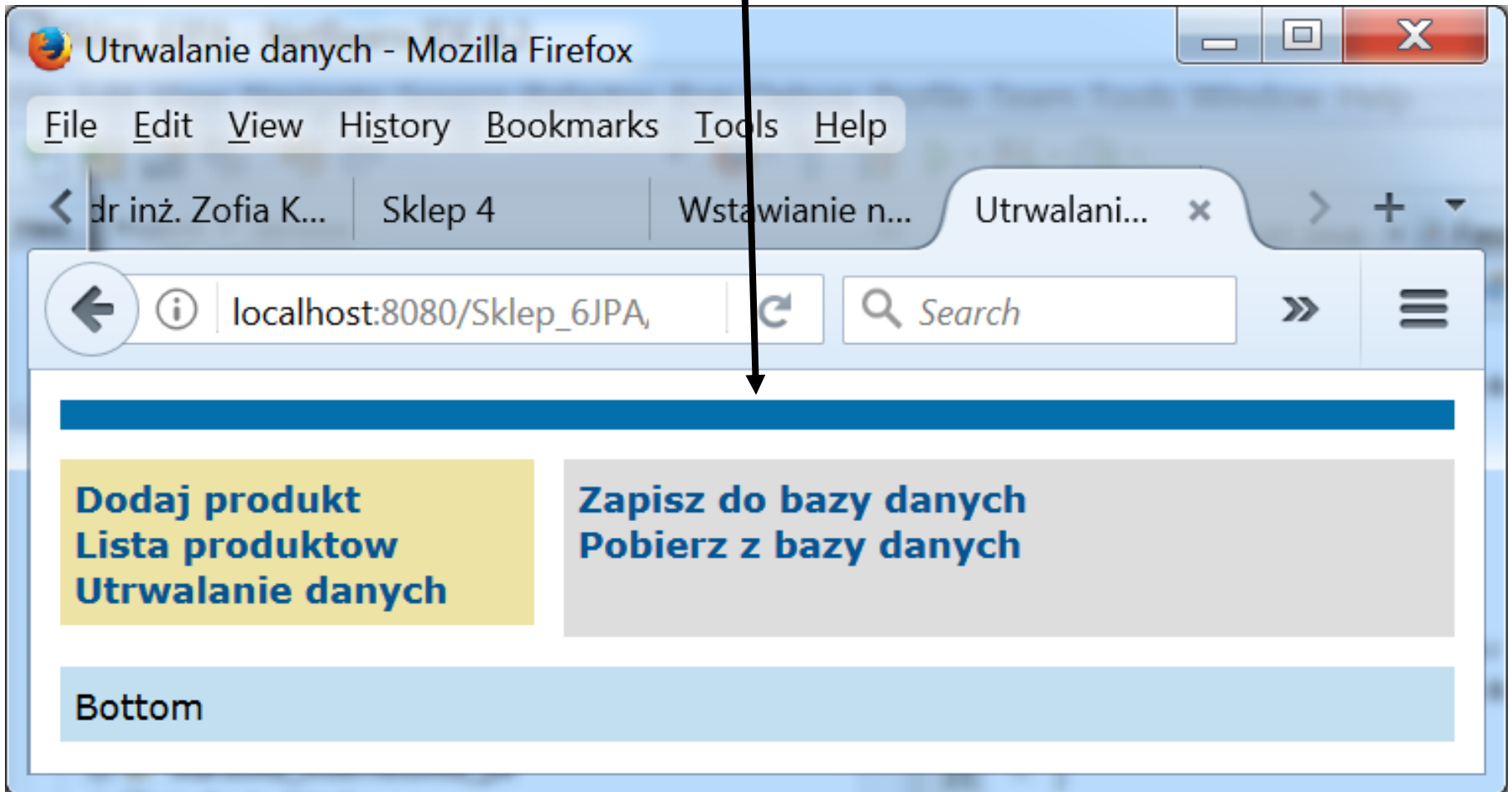
## 35. Prezentacja działania aplikacji – zawartosc tabeli po wstawieniu danych 3 produktów



The screenshot shows a Mozilla Firefox browser window titled "Lista produktow - Mozilla Firefox". The address bar displays "localhost:8080/Sklep\_6JPA/faces/wa". The page content includes a yellow sidebar with navigation links: "Dodaj produkt", "Lista produktow", and "Utrwalanie danych". The main area features a table with 6 columns: "Id produktu", "Nazwa produktu", "Cena netto produktu", "Promocja produktu", "Data produkcji", and "Cena brutto". The table contains three rows of product data. Below the table is a "Powrot" button. The footer of the page contains the text "Bottom".

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto
0	Produkt1	120.0	12	Sat Mar 11 01:00:00 CET 2017	105.6
0	Produkt2	130.0	13	Sat Mar 11 01:00:00 CET 2017	113.1
0	Produkt3	140.0	14	Sat Mar 11 01:00:00 CET 2017	120.4

## 36. Zapis do bazy danych



## 37. Prezentacja bazy danych – należy kliknąć prawym klawiszem myszy na tabelę **Produkt1** i wybrać **View data**.

The screenshot shows the NetBeans IDE 8.2 interface. On the left, the 'Services' pane displays a tree view of the 'Produkt' database schema, including tables like 'PRODUKT1', 'SEQUENCE', and 'SQL170325233539710'. The 'PRODUKT1' table is selected. The main editor shows an SQL query: `SELECT * FROM PRODUKT.PRODUKT1 FETCH FIRST 100 ROWS ONLY;`. Below the query, the results are displayed in a table with columns: #, ID, CENA, DATA\_PRODUKCJI, NAZWA, and PROMOCJA. The results show three rows of data.

#	ID	CENA	DATA_PRODUKCJI	NAZWA	PROMOCJA
1	2	130.0	2017-03-11	Produkt2	13
2	1	120.0	2017-03-11	Produkt1	12
3	3	140.0	2017-03-11	Produkt3	14

The bottom status bar shows the current view is '1:1' and the cursor is in 'INS' mode. The output pane at the bottom right shows the execution of SQL 3.



37 (cd) Tablica SEQUENCE zawiera wartość wykorzystywaną przy generowaniu klucza głównego, gdy adnotacja przy id w klasie Produkt1 jest:

**@Id**

**@GeneratedValue(strategy = GenerationType.AUTO)**

private Long id;

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Files Projects Services

jdbc:derby://localhost:1527/Produkt [F

PRODUKT

- Tables
  - PRODUKT1
    - ID
    - CENA
    - DATA\_PRODUKCJI
    - NAZWA
    - PROMOCJA
  - Indexes
  - Foreign Keys
  - SEQUENCE**
    - SEQ\_NAME
    - SEQ\_COUNT
  - Indexes
    - SQL170325233539710
  - Foreign Keys
- Views
- Procedures

Navigator

- .right\_content
- Elements

SQL 4 [jdbc:derby://localhost:15...]

Connection: jdbc:derby://localhost:1527/Produkt [Produkt on PR...

```
1 SELECT * FROM PRODUKT."SEQUENCE" FETCH FIRST 100 ROWS ONLY;
```

2

SELECT \* FROM PRODUKT."SE...

Max. rows: 100 | Fetched Rows: 1 | Matching Rows:

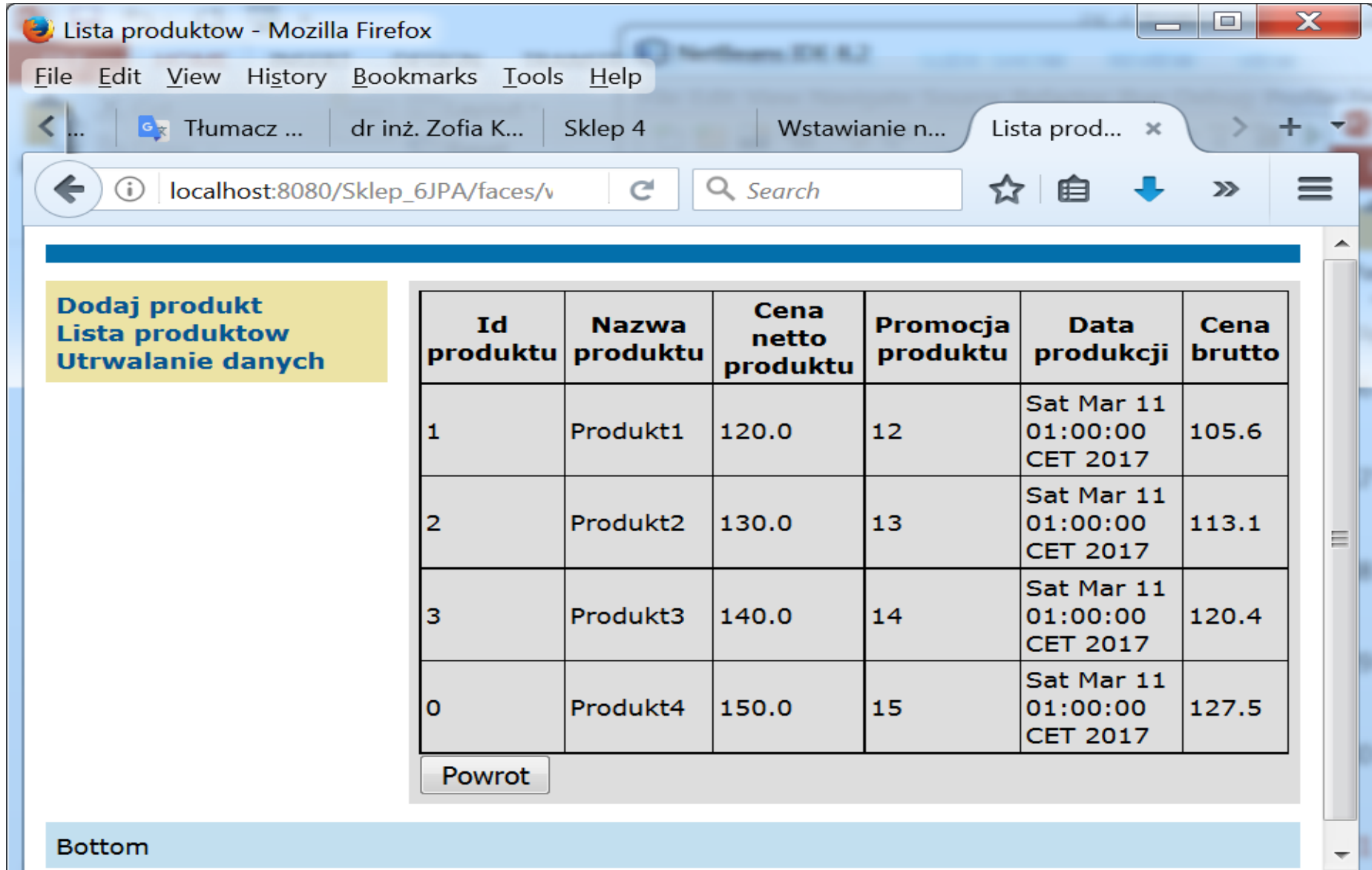
#	SEQ_NAME	SEQ_COUNT
1	SEQ_GEN	50

Output

1.1 Sklep\_6JPA (run) SQL 1 execution SQL 2 execution SQL 3 execution

1:1 INS

38. Prezentacja działania aplikacji – zawartość tabeli po wstawieniu danych 4 produktów (przy zapisie do bazy danych następuje zmiana wartości id zapisanych 3 produktów)



Lista produktow - Mozilla Firefox

File Edit View History Bookmarks Tools Help

localhost:8080/Sklep\_6JPA/faces/v

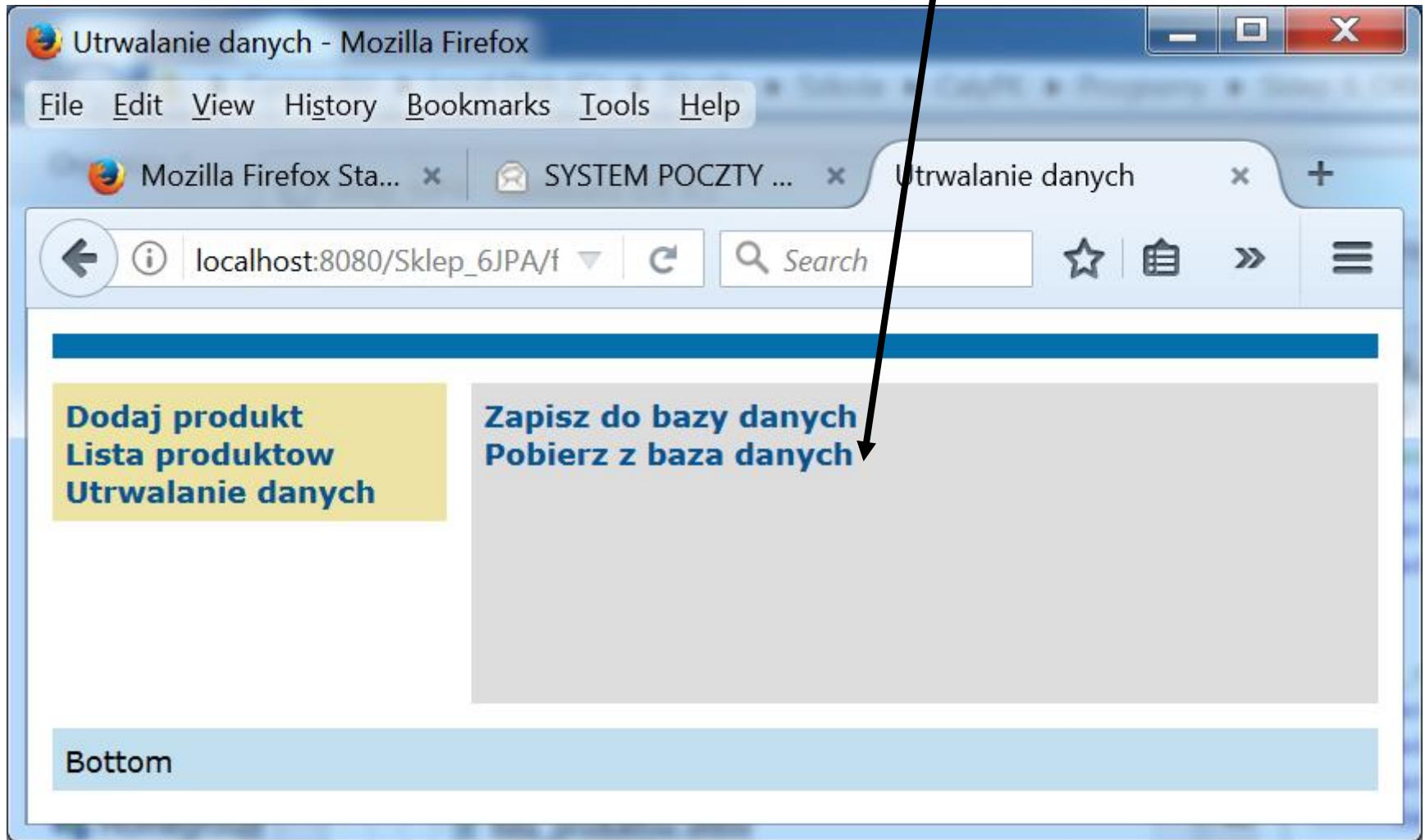
**Dodaj produkt**  
**Lista produktow**  
**Utrwalanie danych**

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto
1	Produkt1	120.0	12	Sat Mar 11 01:00:00 CET 2017	105.6
2	Produkt2	130.0	13	Sat Mar 11 01:00:00 CET 2017	113.1
3	Produkt3	140.0	14	Sat Mar 11 01:00:00 CET 2017	120.4
0	Produkt4	150.0	15	Sat Mar 11 01:00:00 CET 2017	127.5

Powrot

Bottom

## 39. Odczyt z bazy danych



## 40. Prezentacja działania aplikacji – zawartość tabeli po odczycie z bazy danych

Lista produktów - Mozilla Firefox

File Edit View History Bookmarks Tools Help

... Tłumacz ... dr inż. Zofia K... Sklep 4 Wstawianie n... Lista prod... x

localhost:8080/Sklep\_6JPA/faces/v Search

**Dodaj produkt**  
**Lista produktów**  
**Utrwalanie danych**

Id produktu	Nazwa produktu	Cena netto produktu	Promocja produktu	Data produkcji	Cena brutto
2	Produkt2	130.0	13	Sat Mar 11 01:00:00 CET 2017	113.1
1	Produkt1	120.0	12	Sat Mar 11 01:00:00 CET 2017	105.6
3	Produkt3	140.0	14	Sat Mar 11 01:00:00 CET 2017	120.4

Powrot

Bottom