

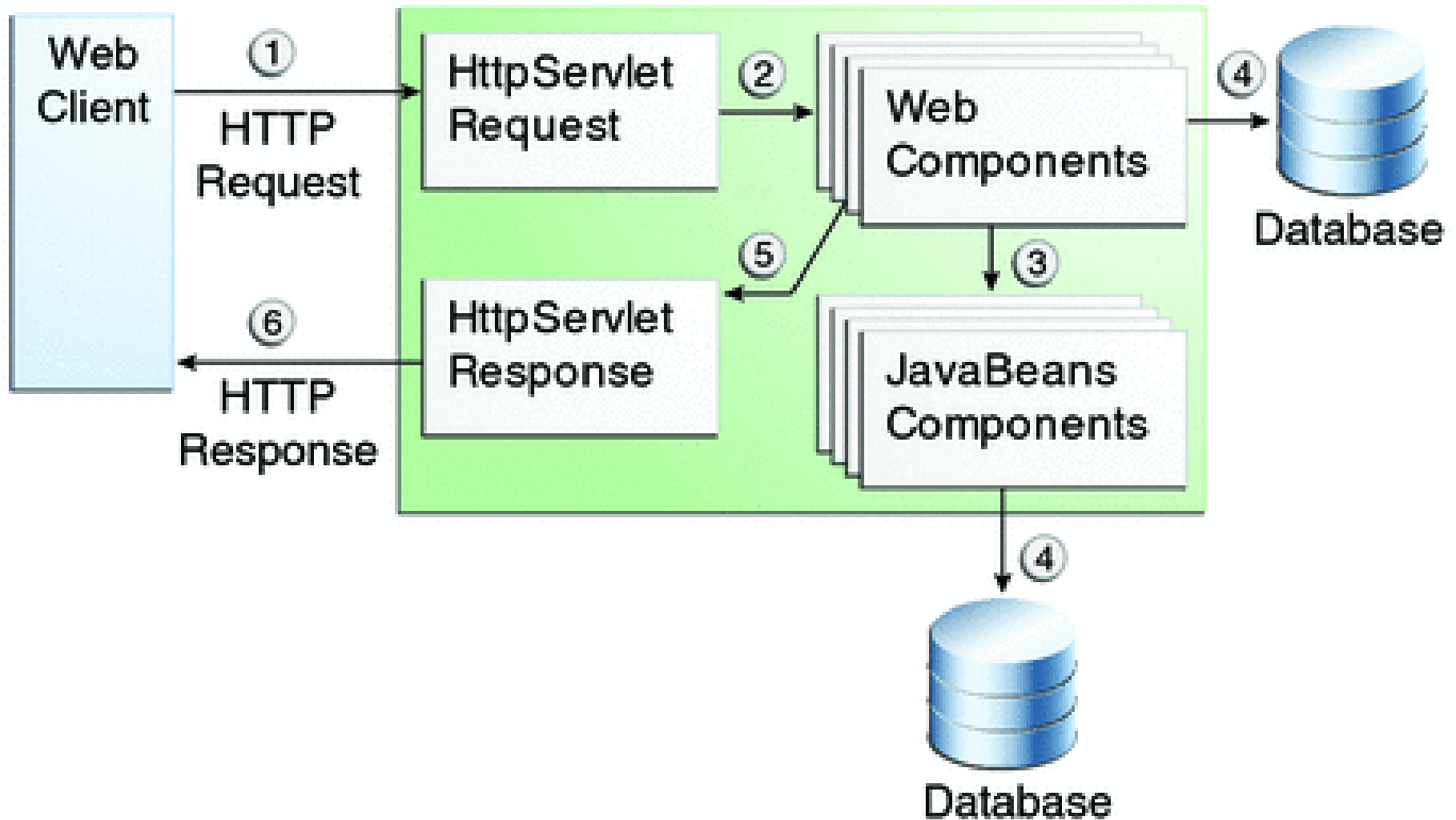
Wprowadzenie do technologii JavaServer Faces 2.1

na podstawie

<http://docs.oracle.com/javaee/6/tutorial/doc/>

**Aplikacja internetowa tworzona na
podstawie bazy danych.**

Obsługa żądania w aplikacji internetowej JAVA

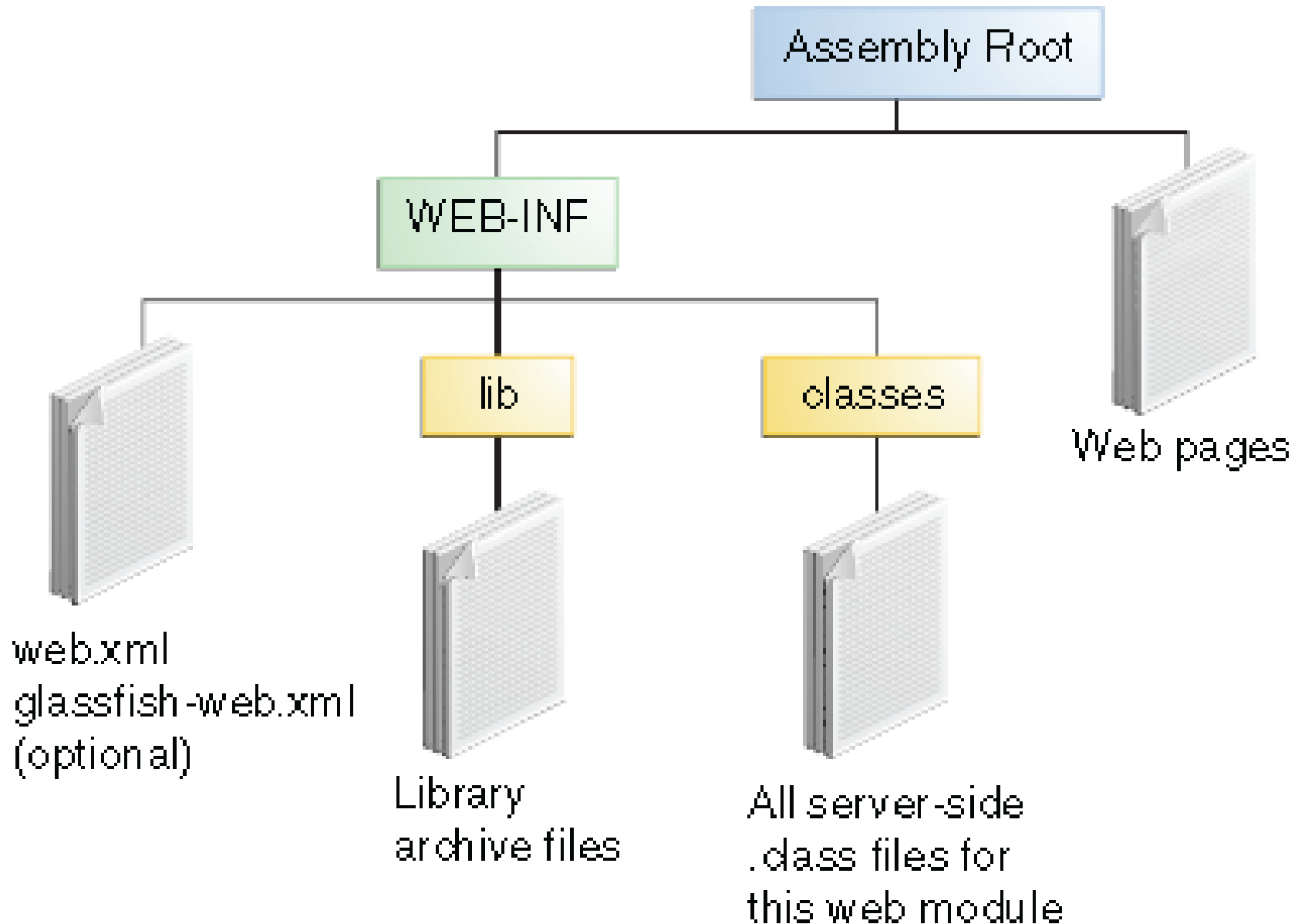


Cykl życia aplikacji internetowej Java EE

Proces tworzenia, wdrażania i wykonywania aplikacji internetowej:

1. Opracowanie kodu komponentu WWW.
2. Rozwijanie deskryptora wdrożenia aplikacji internetowej, jeśli to konieczne (web.xml).
3. Kompilacja komponentów aplikacji internetowej i klas pomocniczych.
4. Opcjonalnie: spakowanie aplikacji w pakiet umożliwiający uruchomienie aplikacji internetowej.
5. Połączenie aplikacji z kontenerem internetowym (deploy).
6. Uruchomienie aplikacji internetowej - dostęp do adresu URL, do którego odwołuje się aplikacja internetowa.

Struktura modułu internetowego



Co zawiera aplikacja Java Server Faces?

Web pages: pliki xhtml, css, js

WEB-INF/classes: katalog, który zawiera po stronie serwera klasy: serwlety, pliki komponentów EJB, klasy użytkowe i komponenty JavaBeans

WEB-INF/lib: katalog, który zawiera pliki JAR z komponentami EJB oraz archiwa bibliotek używanych przez klasy aplikacji

WEB-INF: deskryptory wdrażania - web.xml i glassfish-web.xml (pliki opisujące instalację aplikacji)

1.2. Pięciorzędowy model logicznego rozdzielania zadań (wg. D.Alur, J.Crupi, D. Malks, Core J2EE. Wzorce projektowe.)



Przykład – aplikacja bazodanowa JSF

<http://netbeans.org/kb/docs/web/jsf20-crud.html>

1. Tworzenie pustej bazy danych (1)

Zakładanie pustej bazy danych w systemie baz danych Derby: Database Name: Katalogksiazek, User Name: Katalogksiazek, Password: Katalogksiazek

The screenshot displays the NetBeans IDE 7.2 interface. The 'Databases' tree view on the left shows a list of databases, including several Derby databases. A context menu is open over the 'Katalogksiazek' database, with the 'Create Database...' option selected. The 'Create Java DB Database' dialog box is open in the foreground, showing the following fields:

- Database Name: Katalogksiazek
- User Name: Katalogksiazek
- Password: [masked with dots]
- Confirm Password: [masked with dots]
- Database Location: C:\Users\kruczkiewicz\.netbeans-derby

The dialog box also features an 'OK' button, a 'Cancel' button, and a 'Properties...' button next to the Database Location field. The background shows the NetBeans IDE interface with the 'Databases' tree view on the left and the 'NetBeans IDE' start page on the right.

1. Połączenie z pustą bazą danych (2)

The screenshot displays the NetBeans IDE 7.2 interface. The 'Services' tab is active in the left-hand 'Projects' pane, showing a list of database connections. A context menu is open over the selected connection 'jdbc:derby://localhost:1527/Katalog_ksiazek1 [Katalog_ksiazek1 on KATALOG_KSIAZEK1]', with the 'Connect...' option highlighted. The main editor area shows the 'Start Page' of the Java ME SDK, and the 'Output' window displays the message 'Serwer sieciowy Apache Derby - 10.4.2.1 - (706043) uruchomiony'. The 'HTTP Server Monitor' window is also visible at the bottom right.

NetBeans IDE 7.2

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

Search (Ctrl+I)

localhost

Projects Files Services Favorites

Databases

- MySQL Server at localhost:3306 [root] (disconnected)
- Java DB
- Drivers
- jdbc:derby://localhost:1527/alarmsystem [alarm on ALARM]
- jdbc:derby://localhost:1527/consult [consult on CONSULT]
- jdbc:derby://localhost:1527/JsfJpa [JsfJpa on JSFJPA]
- jdbc:derby://localhost:1527/Katalog [katalog on KATALOG]
- jdbc:derby://localhost:1527/Katalog_Klientow [Katalog_Klientow on KATALOG_KLIENTOW]
- jdbc:derby://localhost:1527/Katalog_ksiazek [Katalog_ksiazek on KATALOG_KSIAZEK]
- jdbc:derby://localhost:1527/Katalog_ksiazek1 [Katalog_ksiazek1 on KATALOG_KSIAZEK1]
- jdbc:derby://localhost:1527/Katalog_ksiazek2 [Katalog_ksiazek2 on KATALOG_KSIAZEK2]
- jdbc:derby://localhost:1527/Katalog_ksiazek3 [Katalog_ksiazek3 on KATALOG_KSIAZEK3]
- jdbc:derby://localhost:1527/Katalog_ksiazek4 [Katalog_ksiazek4 on KATALOG_KSIAZEK4]
- jdbc:derby://localhost:1527/Katalog_ksiazek5 [Katalog_ksiazek5 on KATALOG_KSIAZEK5]
- jdbc:derby://localhost:1527/Katalog_ksiazek6 [Katalog_ksiazek6 on KATALOG_KSIAZEK6]
- jdbc:derby://localhost:1527/Katalog_ksiazek7 [Katalog_ksiazek7 on KATALOG_KSIAZEK7]
- jdbc:derby://localhost:1527/Katalog_ksiazek8 [Katalog_ksiazek8 on KATALOG_KSIAZEK8]
- jdbc:derby://localhost:1527/Katalog_ksiazek9 [Katalog_ksiazek9 on KATALOG_KSIAZEK9]
- jdbc:derby://localhost:1527/Katalog_ksiazek10 [Katalog_ksiazek10 on KATALOG_KSIAZEK10]
- jdbc:derby://localhost:1527/Katalog_ksiazek11 [Katalog_ksiazek11 on KATALOG_KSIAZEK11]
- jdbc:derby://localhost:1527/Katalog_ksiazek12 [Katalog_ksiazek12 on KATALOG_KSIAZEK12]
- jdbc:derby://localhost:1527/Katalog_ksiazek13 [Katalog_ksiazek13 on KATALOG_KSIAZEK13]
- jdbc:derby://localhost:1527/Katalog_ksiazek14 [Katalog_ksiazek14 on KATALOG_KSIAZEK14]
- jdbc:derby://localhost:1527/Katalog_ksiazek15 [Katalog_ksiazek15 on KATALOG_KSIAZEK15]
- jdbc:derby://localhost:1527/Katalog_ksiazek16 [Katalog_ksiazek16 on KATALOG_KSIAZEK16]
- jdbc:derby://localhost:1527/Katalog_ksiazek17 [Katalog_ksiazek17 on KATALOG_KSIAZEK17]
- jdbc:derby://localhost:1527/Katalog_ksiazek18 [Katalog_ksiazek18 on KATALOG_KSIAZEK18]
- jdbc:derby://localhost:1527/Katalog_ksiazek19 [Katalog_ksiazek19 on KATALOG_KSIAZEK19]
- jdbc:derby://localhost:1527/Katalog_ksiazek20 [Katalog_ksiazek20 on KATALOG_KSIAZEK20]
- jdbc:derby://localhost:1527/Katalog_ksiazek21 [Katalog_ksiazek21 on KATALOG_KSIAZEK21]
- jdbc:derby://localhost:1527/Katalog_ksiazek22 [Katalog_ksiazek22 on KATALOG_KSIAZEK22]
- jdbc:derby://localhost:1527/Katalog_ksiazek23 [Katalog_ksiazek23 on KATALOG_KSIAZEK23]
- jdbc:derby://localhost:1527/Katalog_ksiazek24 [Katalog_ksiazek24 on KATALOG_KSIAZEK24]
- jdbc:derby://localhost:1527/Katalog_ksiazek25 [Katalog_ksiazek25 on KATALOG_KSIAZEK25]
- jdbc:derby://localhost:1527/Katalog_ksiazek26 [Katalog_ksiazek26 on KATALOG_KSIAZEK26]
- jdbc:derby://localhost:1527/Katalog_ksiazek27 [Katalog_ksiazek27 on KATALOG_KSIAZEK27]
- jdbc:derby://localhost:1527/Katalog_ksiazek28 [Katalog_ksiazek28 on KATALOG_KSIAZEK28]
- jdbc:derby://localhost:1527/Katalog_ksiazek29 [Katalog_ksiazek29 on KATALOG_KSIAZEK29]
- jdbc:derby://localhost:1527/Katalog_ksiazek30 [Katalog_ksiazek30 on KATALOG_KSIAZEK30]
- jdbc:derby://localhost:1527/Katalog_ksiazek31 [Katalog_ksiazek31 on KATALOG_KSIAZEK31]
- jdbc:derby://localhost:1527/Katalog_ksiazek32 [Katalog_ksiazek32 on KATALOG_KSIAZEK32]
- jdbc:derby://localhost:1527/Katalog_ksiazek33 [Katalog_ksiazek33 on KATALOG_KSIAZEK33]
- jdbc:derby://localhost:1527/Katalog_ksiazek34 [Katalog_ksiazek34 on KATALOG_KSIAZEK34]
- jdbc:derby://localhost:1527/Katalog_ksiazek35 [Katalog_ksiazek35 on KATALOG_KSIAZEK35]
- jdbc:derby://localhost:1527/Katalog_ksiazek36 [Katalog_ksiazek36 on KATALOG_KSIAZEK36]
- jdbc:derby://localhost:1527/Katalog_ksiazek37 [Katalog_ksiazek37 on KATALOG_KSIAZEK37]
- jdbc:derby://localhost:1527/Katalog_ksiazek38 [Katalog_ksiazek38 on KATALOG_KSIAZEK38]
- jdbc:derby://localhost:1527/Katalog_ksiazek39 [Katalog_ksiazek39 on KATALOG_KSIAZEK39]
- jdbc:derby://localhost:1527/Katalog_ksiazek40 [Katalog_ksiazek40 on KATALOG_KSIAZEK40]
- jdbc:derby://localhost:1527/Katalog_ksiazek41 [Katalog_ksiazek41 on KATALOG_KSIAZEK41]
- jdbc:derby://localhost:1527/Katalog_ksiazek42 [Katalog_ksiazek42 on KATALOG_KSIAZEK42]
- jdbc:derby://localhost:1527/Katalog_ksiazek43 [Katalog_ksiazek43 on KATALOG_KSIAZEK43]
- jdbc:derby://localhost:1527/Katalog_ksiazek44 [Katalog_ksiazek44 on KATALOG_KSIAZEK44]
- jdbc:derby://localhost:1527/Katalog_ksiazek45 [Katalog_ksiazek45 on KATALOG_KSIAZEK45]
- jdbc:derby://localhost:1527/Katalog_ksiazek46 [Katalog_ksiazek46 on KATALOG_KSIAZEK46]
- jdbc:derby://localhost:1527/Katalog_ksiazek47 [Katalog_ksiazek47 on KATALOG_KSIAZEK47]
- jdbc:derby://localhost:1527/Katalog_ksiazek48 [Katalog_ksiazek48 on KATALOG_KSIAZEK48]
- jdbc:derby://localhost:1527/Katalog_ksiazek49 [Katalog_ksiazek49 on KATALOG_KSIAZEK49]
- jdbc:derby://localhost:1527/Katalog_ksiazek50 [Katalog_ksiazek50 on KATALOG_KSIAZEK50]
- jdbc:derby://localhost:1527/Katalog_ksiazek51 [Katalog_ksiazek51 on KATALOG_KSIAZEK51]
- jdbc:derby://localhost:1527/Katalog_ksiazek52 [Katalog_ksiazek52 on KATALOG_KSIAZEK52]
- jdbc:derby://localhost:1527/Katalog_ksiazek53 [Katalog_ksiazek53 on KATALOG_KSIAZEK53]
- jdbc:derby://localhost:1527/Katalog_ksiazek54 [Katalog_ksiazek54 on KATALOG_KSIAZEK54]
- jdbc:derby://localhost:1527/Katalog_ksiazek55 [Katalog_ksiazek55 on KATALOG_KSIAZEK55]
- jdbc:derby://localhost:1527/Katalog_ksiazek56 [Katalog_ksiazek56 on KATALOG_KSIAZEK56]
- jdbc:derby://localhost:1527/Katalog_ksiazek57 [Katalog_ksiazek57 on KATALOG_KSIAZEK57]
- jdbc:derby://localhost:1527/Katalog_ksiazek58 [Katalog_ksiazek58 on KATALOG_KSIAZEK58]
- jdbc:derby://localhost:1527/Katalog_ksiazek59 [Katalog_ksiazek59 on KATALOG_KSIAZEK59]
- jdbc:derby://localhost:1527/Katalog_ksiazek60 [Katalog_ksiazek60 on KATALOG_KSIAZEK60]
- jdbc:derby://localhost:1527/Katalog_ksiazek61 [Katalog_ksiazek61 on KATALOG_KSIAZEK61]
- jdbc:derby://localhost:1527/Katalog_ksiazek62 [Katalog_ksiazek62 on KATALOG_KSIAZEK62]
- jdbc:derby://localhost:1527/Katalog_ksiazek63 [Katalog_ksiazek63 on KATALOG_KSIAZEK63]
- jdbc:derby://localhost:1527/Katalog_ksiazek64 [Katalog_ksiazek64 on KATALOG_KSIAZEK64]
- jdbc:derby://localhost:1527/Katalog_ksiazek65 [Katalog_ksiazek65 on KATALOG_KSIAZEK65]
- jdbc:derby://localhost:1527/Katalog_ksiazek66 [Katalog_ksiazek66 on KATALOG_KSIAZEK66]
- jdbc:derby://localhost:1527/Katalog_ksiazek67 [Katalog_ksiazek67 on KATALOG_KSIAZEK67]
- jdbc:derby://localhost:1527/Katalog_ksiazek68 [Katalog_ksiazek68 on KATALOG_KSIAZEK68]
- jdbc:derby://localhost:1527/Katalog_ksiazek69 [Katalog_ksiazek69 on KATALOG_KSIAZEK69]
- jdbc:derby://localhost:1527/Katalog_ksiazek70 [Katalog_ksiazek70 on KATALOG_KSIAZEK70]
- jdbc:derby://localhost:1527/Katalog_ksiazek71 [Katalog_ksiazek71 on KATALOG_KSIAZEK71]
- jdbc:derby://localhost:1527/Katalog_ksiazek72 [Katalog_ksiazek72 on KATALOG_KSIAZEK72]
- jdbc:derby://localhost:1527/Katalog_ksiazek73 [Katalog_ksiazek73 on KATALOG_KSIAZEK73]
- jdbc:derby://localhost:1527/Katalog_ksiazek74 [Katalog_ksiazek74 on KATALOG_KSIAZEK74]
- jdbc:derby://localhost:1527/Katalog_ksiazek75 [Katalog_ksiazek75 on KATALOG_KSIAZEK75]
- jdbc:derby://localhost:1527/Katalog_ksiazek76 [Katalog_ksiazek76 on KATALOG_KSIAZEK76]
- jdbc:derby://localhost:1527/Katalog_ksiazek77 [Katalog_ksiazek77 on KATALOG_KSIAZEK77]
- jdbc:derby://localhost:1527/Katalog_ksiazek78 [Katalog_ksiazek78 on KATALOG_KSIAZEK78]
- jdbc:derby://localhost:1527/Katalog_ksiazek79 [Katalog_ksiazek79 on KATALOG_KSIAZEK79]
- jdbc:derby://localhost:1527/Katalog_ksiazek80 [Katalog_ksiazek80 on KATALOG_KSIAZEK80]
- jdbc:derby://localhost:1527/Katalog_ksiazek81 [Katalog_ksiazek81 on KATALOG_KSIAZEK81]
- jdbc:derby://localhost:1527/Katalog_ksiazek82 [Katalog_ksiazek82 on KATALOG_KSIAZEK82]
- jdbc:derby://localhost:1527/Katalog_ksiazek83 [Katalog_ksiazek83 on KATALOG_KSIAZEK83]
- jdbc:derby://localhost:1527/Katalog_ksiazek84 [Katalog_ksiazek84 on KATALOG_KSIAZEK84]
- jdbc:derby://localhost:1527/Katalog_ksiazek85 [Katalog_ksiazek85 on KATALOG_KSIAZEK85]
- jdbc:derby://localhost:1527/Katalog_ksiazek86 [Katalog_ksiazek86 on KATALOG_KSIAZEK86]
- jdbc:derby://localhost:1527/Katalog_ksiazek87 [Katalog_ksiazek87 on KATALOG_KSIAZEK87]
- jdbc:derby://localhost:1527/Katalog_ksiazek88 [Katalog_ksiazek88 on KATALOG_KSIAZEK88]
- jdbc:derby://localhost:1527/Katalog_ksiazek89 [Katalog_ksiazek89 on KATALOG_KSIAZEK89]
- jdbc:derby://localhost:1527/Katalog_ksiazek90 [Katalog_ksiazek90 on KATALOG_KSIAZEK90]
- jdbc:derby://localhost:1527/Katalog_ksiazek91 [Katalog_ksiazek91 on KATALOG_KSIAZEK91]
- jdbc:derby://localhost:1527/Katalog_ksiazek92 [Katalog_ksiazek92 on KATALOG_KSIAZEK92]
- jdbc:derby://localhost:1527/Katalog_ksiazek93 [Katalog_ksiazek93 on KATALOG_KSIAZEK93]
- jdbc:derby://localhost:1527/Katalog_ksiazek94 [Katalog_ksiazek94 on KATALOG_KSIAZEK94]
- jdbc:derby://localhost:1527/Katalog_ksiazek95 [Katalog_ksiazek95 on KATALOG_KSIAZEK95]
- jdbc:derby://localhost:1527/Katalog_ksiazek96 [Katalog_ksiazek96 on KATALOG_KSIAZEK96]
- jdbc:derby://localhost:1527/Katalog_ksiazek97 [Katalog_ksiazek97 on KATALOG_KSIAZEK97]
- jdbc:derby://localhost:1527/Katalog_ksiazek98 [Katalog_ksiazek98 on KATALOG_KSIAZEK98]
- jdbc:derby://localhost:1527/Katalog_ksiazek99 [Katalog_ksiazek99 on KATALOG_KSIAZEK99]
- jdbc:derby://localhost:1527/Katalog_ksiazek100 [Katalog_ksiazek100 on KATALOG_KSIAZEK100]

Connect... Disconnect Execute Command... Refresh Delete Rename... Properties

Java ME SDK Start Page Start Page

NetBeansIDE

Learn & Discover My NetBeans

News & Tutorials Blogs

Output - Java DB Database Process

```
DRDA_SecurityInstalled.I
Serwer sieciowy Apache Derby - 10.4.2.1 - (706043) uruchomiony
```

HTTP Server Monitor

Request	Client and Server		Headers	
	Cookies	Session	Context	
All Records				
Current Records				
Saved Records				

Biblioteka6 - Navigator

<No View Available>

4 | INS

1. Połączenie z pustą bazą danych (3)

The screenshot displays the NetBeans IDE 7.2 interface. The main workspace is divided into several panes:

- Projects:** A tree view on the left showing database connections. The connection `jdbc:derby://localhost:1527/Katalogksiazek [Katalogksiazek on KATALOGKSZIAZEK]` is selected and highlighted in blue.
- Output - Java DB Database Process:** A window showing the server startup logs:

```
DRDA_SecurityInstalled.I
Serwer sieciowy Apache Derby - 10.4.2.1
```
- HTTP Server Monitor:** A window at the bottom right showing headers for 'Client and Server' with sub-sections for 'Session', 'Context', 'Request', and 'Cookies'.

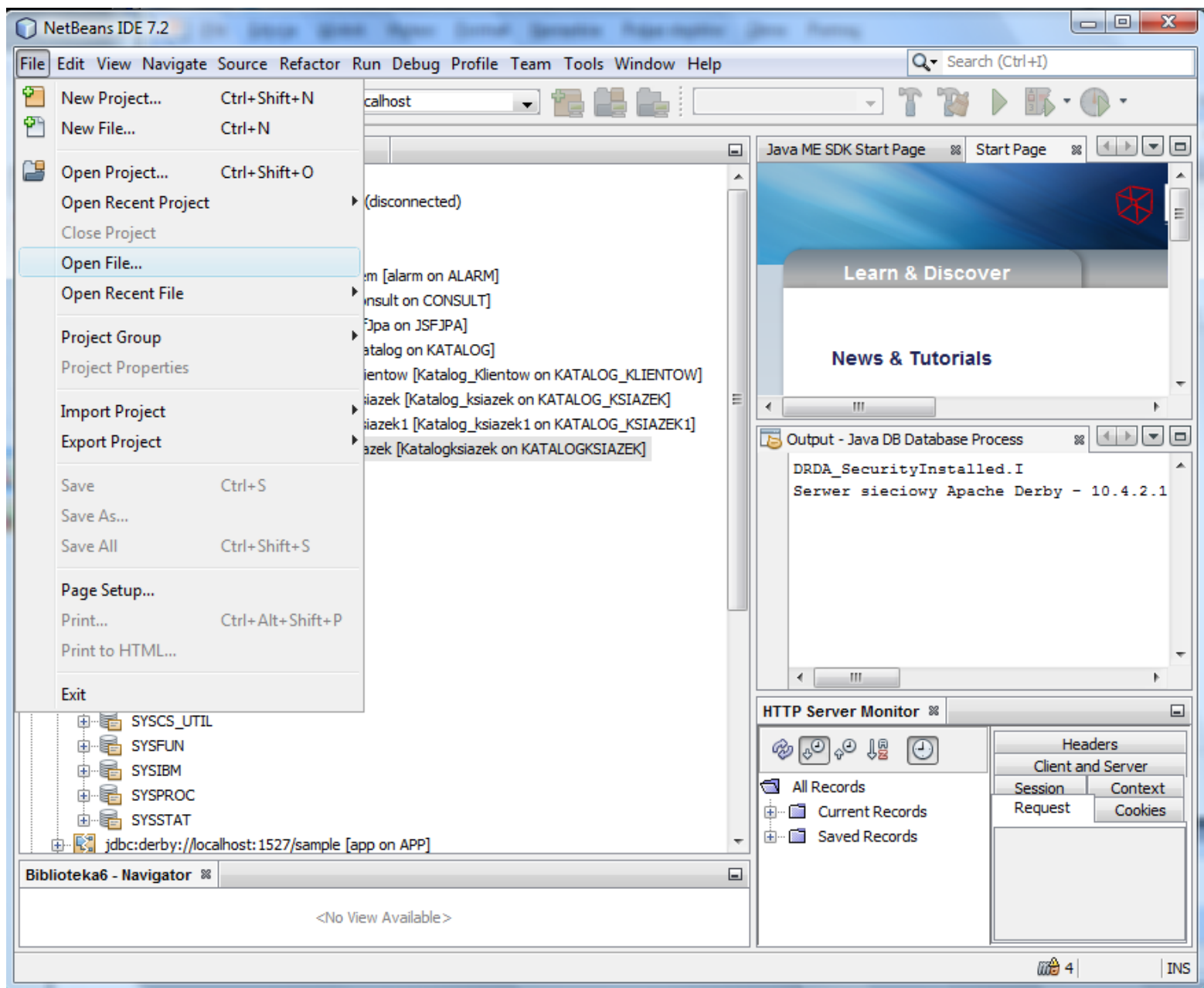
The bottom status bar shows the page number 4 and the text 'INS'.

Wykonanie tabel w bazie danych – skrypt typu sql (1)

```
CREATE TABLE Tytul_ksiazki (  
  tytul_id INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY,  
  tytul VARCHAR(50) NOT NULL,  
  autor_nazwisko VARCHAR(50) NOT NULL,  
  autor_imie VARCHAR(50) NOT NULL,  
  ISBN VARCHAR(50) NOT NULL,  
  wydawnictwo VARCHAR(50) NOT NULL,  
  CONSTRAINT tytul_pk PRIMARY KEY ( tytul_id )  
);
```

```
CREATE TABLE Ksiazka (  
  ksiazka_id INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY,  
  numer INTEGER NOT NULL,  
  id_tytul INTEGER NOT NULL  
  CONSTRAINT ksiazka_id PRIMARY KEY ( ksiazka_id ),  
  FOREIGN KEY (id_tytul) REFERENCES Tytul_ksiazki (tytul_id)  
);
```

Wykonanie tabel w bazie danych (2) – otwieranie pliku zawierającego polecenia SQL do tworzenia tabel



Wczytanie skryptu typu sql : File/Open File (3)

The screenshot displays the NetBeans IDE 7.2 interface. The main editor window shows a SQL script with the following content:

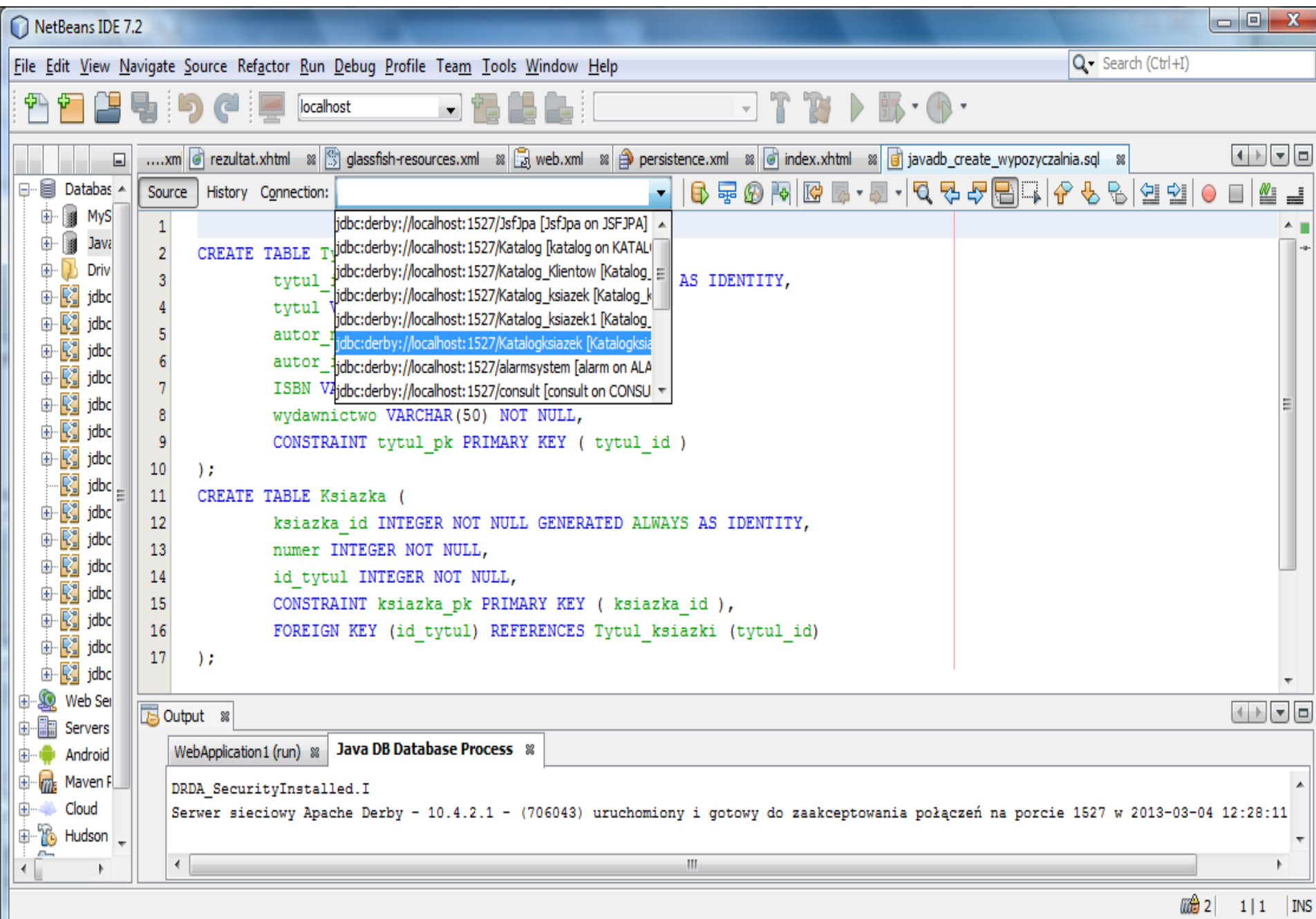
```
1  
2 CREATE TABLE Tytul_książki (  
3     tytul_id INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY,  
4     tytul VARCHAR(50) NOT NULL,  
5     autor_nazwisko VARCHAR(50) NOT NULL,  
6     autor_imie VARCHAR(50) NOT NULL,  
7     ISBN VARCHAR(50) NOT NULL,  
8     wydawnictwo VARCHAR(50) NOT NULL,  
9     CONSTRAINT tytul_pk PRIMARY KEY ( tytul_id )  
10 );  
11 CREATE TABLE Książka (  
12     książka_id INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY,  
13     numer INTEGER NOT NULL,  
14     id_tytul INTEGER NOT NULL,  
15     CONSTRAINT książka_pk PRIMARY KEY ( książka_id ),  
16     FOREIGN KEY (id_tytul) REFERENCES Tytul_książki (tytul_id)  
17 );
```

The left sidebar shows the 'Databases' section with 'Java DB' selected. The 'Output' window at the bottom shows the following message:

```
WebApplication1 (run) Java DB Database Process  
DRDA_SecurityInstalled.I  
Serwer sieciowy Apache Derby - 10.4.2.1 - (706043) uruchomiony i gotowy do zaakce
```

The status bar at the bottom right indicates '2 | 1 | 1 | INS'.

Połączenie skryptu typu sql z bazą danych (4)



The screenshot displays the NetBeans IDE 7.2 interface. The main editor window shows a SQL script with the following content:

```
1 jdbc:derby://localhost:1527/JsfJpa [JsfJpa on JSFJPA]
2 CREATE TABLE Tytul (
3     tytul_id INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY,
4     tytul VARCHAR(255) NOT NULL,
5     autor VARCHAR(255) NOT NULL,
6     isbn VARCHAR(13) NOT NULL,
7     isbn VARCHAR(13) NOT NULL,
8     wydawnictwo VARCHAR(50) NOT NULL,
9     CONSTRAINT tytul_pk PRIMARY KEY ( tytul_id )
10 );
11 CREATE TABLE Ksiazka (
12     ksiazka_id INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY,
13     numer INTEGER NOT NULL,
14     id_tytul INTEGER NOT NULL,
15     CONSTRAINT ksiazka_pk PRIMARY KEY ( ksiazka_id ),
16     FOREIGN KEY (id_tytul) REFERENCES Tytul_ksiazki (tytul_id)
17 );
```

The Output window at the bottom shows the following message:

```
WebApplication1 (run) Java DB Database Process
DRDA_SecurityInstalled.I
Serwer sieciowy Apache Derby - 10.4.2.1 - (706043) uruchomiony i gotowy do zaakceptowania połączeń na porcie 1527 w 2013-03-04 12:28:11
```

Uruchomienie skryptu typu sql (5)

The screenshot shows the NetBeans IDE 7.2 interface. The main editor window displays an SQL script with the following content:

```
1 CREATE TABLE Tytul_książki (  
2     tytul_id INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY,  
3     tytul VARCHAR(50) NOT NULL,  
4     autor_nazwisko VARCHAR(100) NOT NULL,  
5     autor_imie VARCHAR(100) NOT NULL,  
6     isbn VARCHAR(13) NOT NULL,  
7     wydawnictwo VARCHAR(100) NOT NULL,  
8     CONSTRAINT tytul_pk PRIMARY KEY (tytul_id),  
9 );  
10  
11 CREATE TABLE Książka (  
12     książka_id INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY,  
13     numer INTEGER NOT NULL,  
14     id_tytul INTEGER NOT NULL,  
15     CONSTRAINT książka_pk PRIMARY KEY (książka_id),  
16     FOREIGN KEY (id_tytul) REFERENCES Tytul_książki (tytul_id)  
17 );
```

A context menu is open over the script, with the 'Run File' option (Shift+F6) highlighted. Other options include 'Navigate', 'Format' (Alt+Shift+F), 'Run Statement', 'SQL History' (Ctrl+Alt+Shift+H), 'Cut' (Ctrl+X), 'Copy' (Ctrl+C), 'Paste' (Ctrl+V), and 'Select in Projects'.

The 'Output' window at the bottom shows the following log messages:

```
WebApplication1 (run) | Java DB Database Process |  
DRDA_SecurityInstalled.I  
Serwer sieciowy Apache Derby - 10.4.2.1 - (706043) uruchomiony i gotowy do zaakceptowania połączeń na porcie
```

The status bar at the bottom right indicates 2 open files, 4 lines selected, and the 'INS' (Insert) mode.

Utworzenie tabel za pomocą skryptu typu sql (6)

The screenshot displays the NetBeans IDE 7.2 interface. On the left, the 'Projects' pane shows a database project named 'KATALOGKSIAZEK' with a 'Tables' folder containing 'KSIAZKA' and 'TYTUL_KSIAZKI'. The 'KSIAZKA' table has columns 'KSIAZKA_ID', 'NUMER', and 'ID_TYTUL'. The 'TYTUL_KSIAZKI' table has columns 'TYTUL_ID', 'TYTUL', 'AUTOR_NAZWISKO', 'AUTOR_IMIE', 'ISBN', and 'WYDAWNICTWO'. The 'Output' pane at the bottom shows the execution of two SQL scripts. The first script creates the 'Tytul_książki' table with columns: 'tytul_id' (INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY), 'tytul' (VARCHAR(50) NOT NULL), 'autor_nazwisko' (VARCHAR(50) NOT NULL), 'autor_imie' (VARCHAR(50) NOT NULL), 'ISBN' (VARCHAR(50) NOT NULL), and 'wydawnictwo' (VARCHAR(50) NOT NULL). A primary key constraint 'tytul_pk' is defined on 'tytul_id'. The second script creates the 'Książka' table with columns: 'ksiazka_id' (INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY), 'numer' (INTEGER NOT NULL), and 'id_tytul' (INTEGER NOT NULL). A primary key constraint 'ksiazka_pk' is defined on 'ksiazka_id', and a foreign key constraint 'ksiazka_fk' is defined on 'id_tytul' referencing 'tytul_id' in the 'Tytul_książki' table. The execution logs show successful completion for both scripts.

```
1  
2 CREATE TABLE Tytul_książki (  
3     tytul_id INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY,  
4     tytul VARCHAR(50) NOT NULL,  
5     autor_nazwisko VARCHAR(50) NOT NULL,  
6     autor_imie VARCHAR(50) NOT NULL,  
7     ISBN VARCHAR(50) NOT NULL,  
8     wydawnictwo VARCHAR(50) NOT NULL,  
9     CONSTRAINT tytul_pk PRIMARY KEY ( tytul_id )  
10 );  
11 CREATE TABLE Książka (  
12     ksiazka_id INTEGER NOT NULL GENERATED ALWAYS AS IDENTITY,  
13     numer INTEGER NOT NULL,  
14     id_tytul INTEGER NOT NULL,  
15     CONSTRAINT ksiazka_pk PRIMARY KEY ( ksiazka_id ),  
16     FOREIGN KEY (id_tytul) REFERENCES Tytul_książki (tytul_id)  
17 );
```

Output:

```
Java DB Database Process javadb_create_wypozyczalnia.sql execution  
Executed successfully in 0,41 s, 0 rows affected.  
Line 2, column 1  
  
Executed successfully in 0,643 s, 0 rows affected.  
Line 11, column 1  
  
Execution finished after 1,053 s, 0 error(s) occurred.
```


Utworzone tabele oraz możliwe do utworzenia klasy typu Entity

Tabela	Entity	Opis
Tytul_ksiazki	Tytul_ksiazki	Przechowują dane tytułu książki; relacja 1 do wiele z tabelą (encją) Ksiazka
Ksiazka	Ksiazka	Przechowują numer książki Relacja wiele do jeden z tabelą (encją) Tytul_ksiazki

Dodanie danych do tabel za pomocą skryptu typu sql (1)

```
SET SCHEMA Katalogksiazek;
```

```
INSERT INTO Tytul_ksiazki (tytul, autor_nazwisko, autor_imie, ISBN, wydawnictwo)  
  VALUES ('Krzyzacy', 'Sienkiewicz', 'Henryk', '1234567', 'PWN');
```

```
INSERT INTO Ksiazka (numer,id_tytul) VALUES (1,1);
```

Dodanie danych do tabel za pomocą skryptu typu sql (2)

The screenshot displays the NetBeans IDE 7.2 interface. On the left, the Project Explorer shows a database connection to 'jdbc:derby://localhost:1527/Katalogksiazka'. The database schema 'KATALOGKSIAZEK' is expanded, showing tables like 'KSIAZKA' and 'TYTUL_KSIAZKI'. The main editor window shows a SQL script named 'javadb_insert_data_wypożyczalnia.sql' with the following content:

```
1 SET SCHEMA Katalogksiazek;
2
3 INSERT INTO Tytul_ksiazki (tytul, autor_nazwisko, autor_imie, ISBN, wydawnictwo)
4     VALUES ('Kryzacy', 'Sienkiewicz', 'Henryk', '1234567', 'PWN');
5
6 INSERT INTO Ksiazka (numer, id_tytul) VALUES (1,1);
```

A context menu is open over the script, with the 'Run File' option (Shift+F6) highlighted. The Output window at the bottom shows the execution results:

```
WebApplication1 (run) | Java DB Database
Executed successfully in 0,059 s
Line 2, column 1

Executed successfully in 0,04 s, 0 rows affected.
Line 11, column 1

Execution finished after 0,099 s, 0 error(s) occurred.
```

The status bar at the bottom right indicates '2 | 6 | 27 | INS'.

Dodanie danych do tabel za pomocą skryptu typu sql (2)

The screenshot displays the NetBeans IDE 7.2 interface. On the left, the Project Explorer shows a database connection to 'localhost:1527/Katalogksiazek'. The database schema includes tables 'KSIAZKA' and 'TYTUL_KSIAZKI', along with various indexes and foreign keys. The main editor window shows a SQL script named 'javadb_insert_data_wypożyczalnia.sql' with the following content:

```
1 SET SCHEMA Katalogksiazek;  
2  
3 INSERT INTO Tytul_ksiazki (tytul, autor_nazwisko, autor_imie, ISBN, wydawnictwo)  
4     VALUES ('Krzyzacy', 'Sienkiewicz', 'Henryk', '1234567', 'PWN');  
5  
6 INSERT INTO Ksiazka (numer, id_tytul) VALUES (1,1);
```

The Output window at the bottom shows the execution results for the script:

```
Java DB Database Process » javadb_create_wypożyczalnia.sql execution » javadb_insert_data_wypożyczalnia.sql e...  
  
Executed successfully in 0,002 s, 0 rows affected.  
Line 1, column 1  
  
Executed successfully in 0,067 s, 1 rows affected.  
Line 3, column 1  
  
Executed successfully in 0,026 s, 1 rows affected.  
Line 6, column 1
```

The status bar at the bottom right indicates the current page is 2 of 3, and the active window is 'INS'.

Wykonanie zapytania wyświetlającego zawartość wybranej tabeli

The screenshot shows the NetBeans IDE 7.2 interface. On the left, the 'Projects' pane shows a database connection 'jdbc:derby://localhost:1527/Katalogksi...' with a table 'KSIAZKA' selected. A context menu is open over the table, with 'View Data...' selected. The central editor shows the SQL query: `select * from KATALOGKSIAZEK.KSIAZKA`. Below the editor, the results are displayed in a table with the following data:

ZKA_ID	NUMER	ID_TYTUL
3	1	1

The 'Output' window at the bottom shows the execution log:

```
process » javadb_insert_data_wypożyczalnia.sql execution » SQL Command 1  
Executed successfully in 0,109 s.  
Line 1, column 1  
  
Execution finished after 0,109 s, 0 error(s) occurred.
```

Zawartość tabeli Tytul_książki

NetBeans IDE 7.2

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

Search (Ctrl+I)

localhost

Projects Files Ser... Favorit...

jdbc:derby://localhost:1527/Katalogksi

APP

KATALOGKSIAZEK

Tables

KSIAZKA

KSIAZKA_ID

NUMER

ID_TYTUL

Indexes

SQL 130302234036

SQL 130302234036

Foreign Keys

SQL 130302234036

TYTUL_KSIAZKI

TYTUL_ID

TYTUL

AUTOR_NAZWISKO

AUTOR_IMIE

ISBN

WYDAWNICTWO

Indexes

SQL 130302233158

Foreign Keys

Views

Procedures

NULLID

SQLJ

SYS

SYSCAT

SYSCS_DIAG

...ava javadb_insert_data_wypożyczalnia.sql SQL Command 1 SQL Command 2

Source History Connection: j...

```
1 select * from KATALOGKSIAZEK.TYTUL_KSIAZKI
```

select * from KATALOGKSIAZEK.TYTUL_KSIAZKI

Page Size: 20 Total Rows: 1 Page: 1 of 1 Matching Rows:

#	TYTUL_ID	TYTUL	AUTOR_NAZWISKO	AUTOR_IMIE	ISBN	WYDAWNICTWO
1	1	Krzyżacy	Sienkiewicz	Henryk	1234567	PWN

Output

javadb_insert_data_wypożyczalnia.sql execution SQL Command 1 execution SQL Command 2 execution

Executed successfully in 0,109 s.
Line 1, column 1

Execution finished after 0,109 s, 0 error(s) occurred.

4 | 1 | 1 | INS

Zawartość tabeli Ksiazka

NetBeans IDE 7.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Search (Ctrl+I)

localhost

Projects Files Ser... Favorit...

jdbc:derby://localhost:1527/Katalogksi...

APP

KATALOGKSIAZEK

Tables

KSIAZKA

- KSIAZKA_ID
- NUMER
- ID_TYTUL

Indexes

- SQL 130302234036
- SQL 130302234036

Foreign Keys

- SQL 130302234036

TYTUL_KSIAZKI

- TYTUL_ID
- TYTUL
- AUTOR_NAZWISKO
- AUTOR_IMIE
- ISBN
- WYDAWNICTWO

Indexes

- SQL 130302233158

Foreign Keys

Views

Procedures

NULLID

SQLJ

SYS

SYSCAT

SYSCS_DIAG

SQL Command 1

```
select * from KATALOGKSIAZEK.KSIAZKA
```

select * from KATALOGKSIAZEK.KSIAZKA

#	KSIAZKA_ID	NUMER	ID_TYTUL
1	3	1	1

Output

process javadb_insert_data_wypożyczalnia.sql execution SQL Command 1

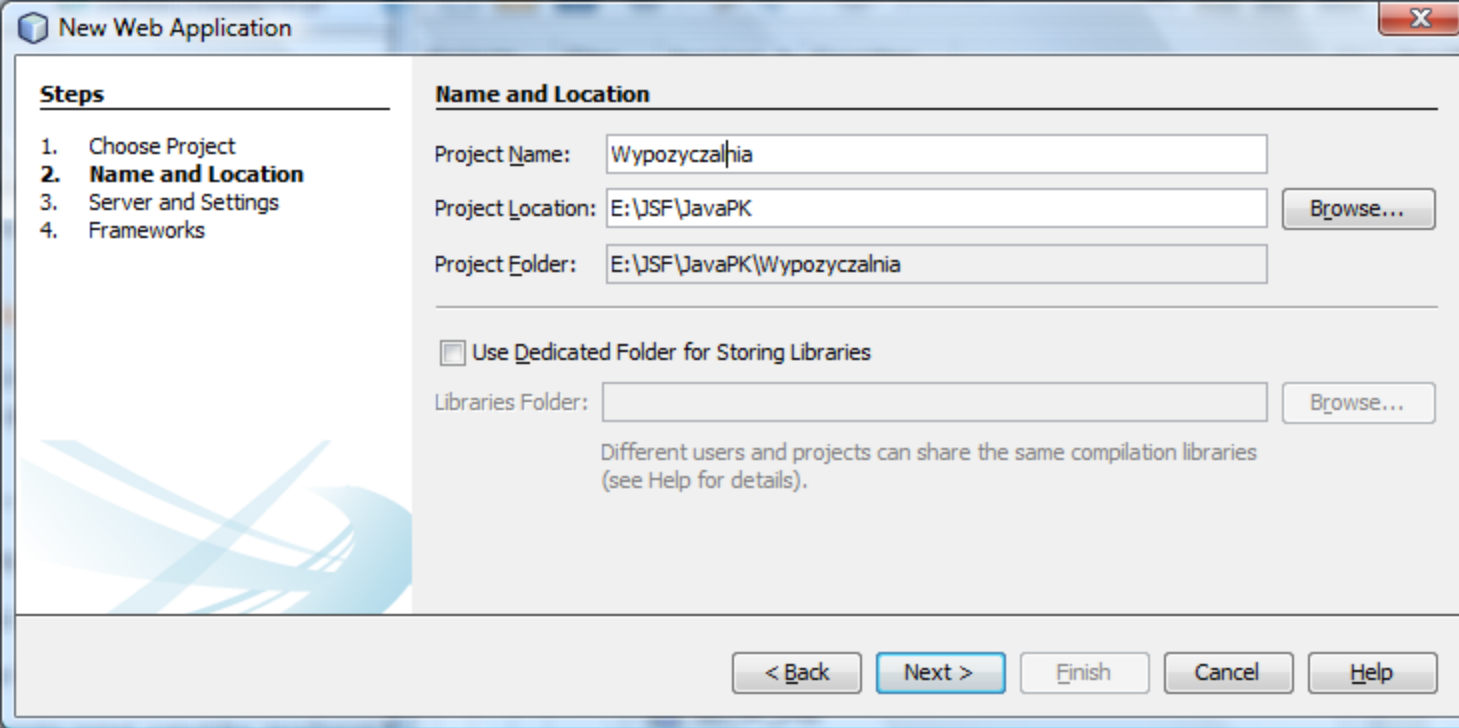
```
Executed successfully in 0,109 s.  
Line 1, column 1  
  
Execution finished after 0,109 s, 0 error(s) occurred.
```

4 | 1 | 1 | INS

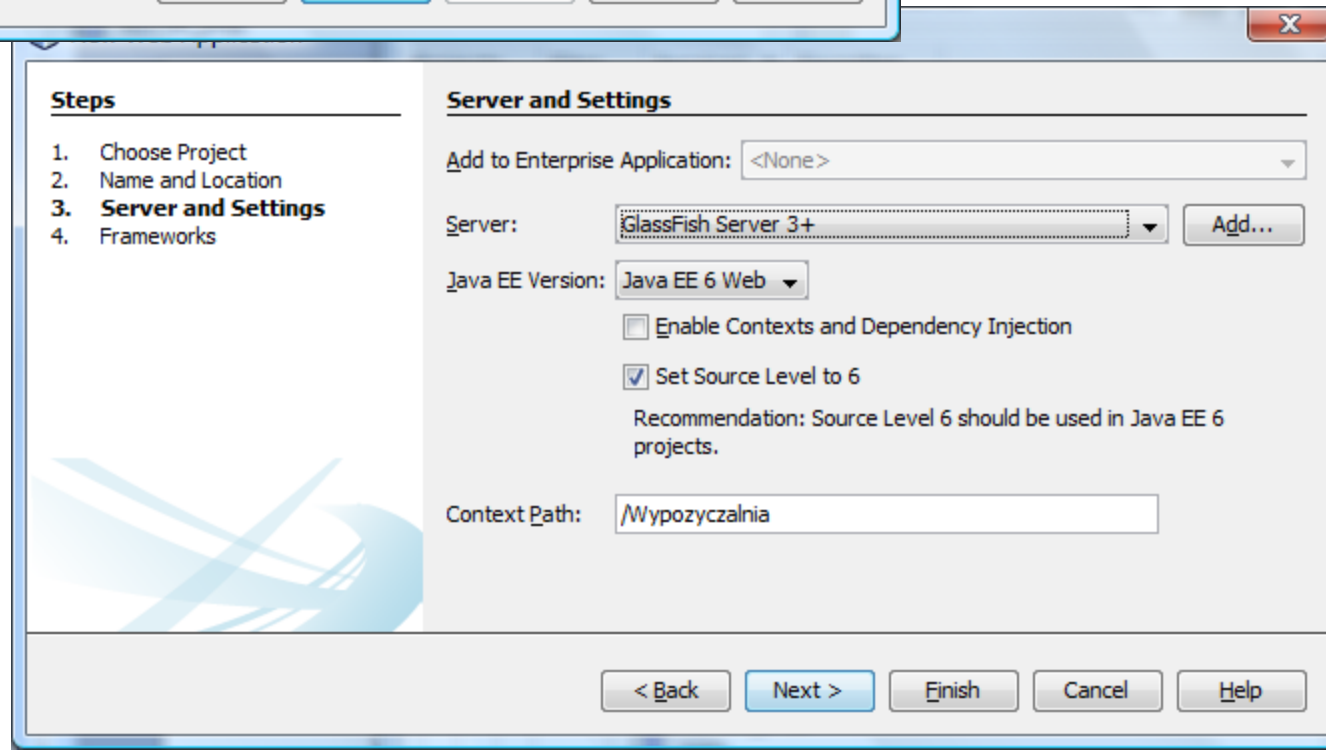
Tworzenie aplikacji typu JavaServer Faces – File/New Project/Java Web/Web Application (1)

The screenshot displays the NetBeans IDE 7.2 interface. The 'File' menu is open, and the 'New Project...' option is selected. The 'New Project' dialog box is in the foreground, showing the 'Choose Project' step. The 'Categories' list includes Java, JavaFX, Java Web, Java EE, Java Card, Java ME, and Maven. The 'Projects' list includes Web Application, Web Application with Existing Sources, and Web Free-Form Application. The 'Web Application' project is selected. The description for 'Web Application' reads: 'Creates an empty Web application in a standard IDE project. A standard project uses an IDE-generated build script to build, run, and debug your project.' The background shows a code editor with the following SQL code:

```
1 SET SCHEMA Katalogsiazek;  
2  
3 INSERT INTO Tytul_ksiazki (tytul, autor_nazwisko, autor_  
4 VALUES ('Kryzacy', 'Sienkiewicz', 'Henryk', '123  
5  
6
```

Tworzenie aplikacji internetowej JSF (2) o nazwie Wypozyczalnia



Tworzenie aplikacji internetowej JSF (3)

New Web Application X

Steps

1. Choose Project
2. Name and Location
3. Server and Settings
- 4. Frameworks**

Frameworks

Select the frameworks you want to use in your web application.

- Spring Web MVC
- JavaServer Faces
- Struts 1.3.10
- Hibernate 3.2.5

JavaServer Faces Configuration

Libraries Configuration Components

Server Library: JSF 2.1

Registered Libraries: Search Libraries...

Create New Library

JSF Folder or JAR:

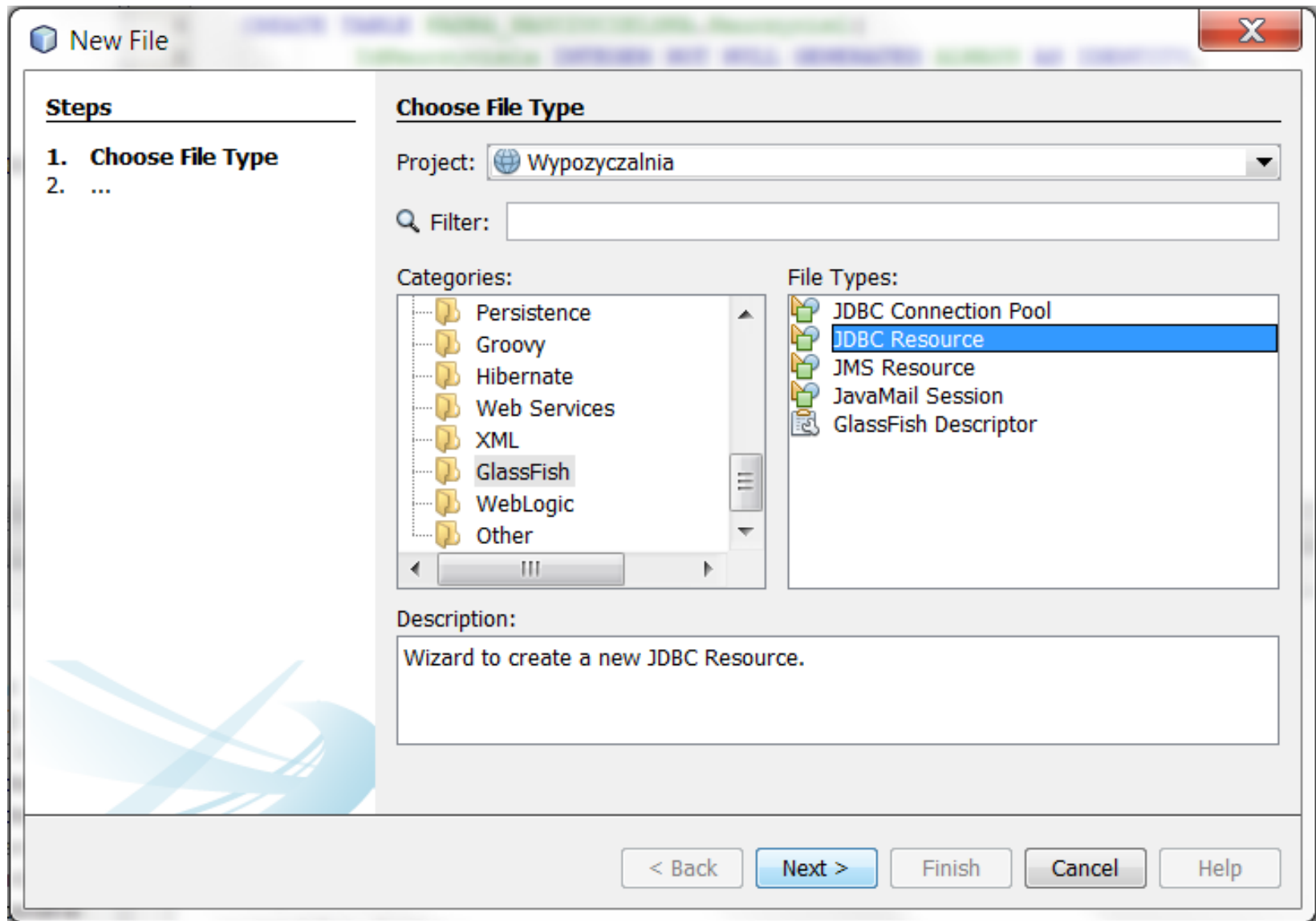
Library Name:

< Back Next > Finish Cancel Help

Wykonanie odniesienia typu JNDI do bazy danych Katalogksiazek (1): **New Data Source**- wybór **New** z listy po kliknięciu prawym klawiszem na nazwę projektu w zakładce **Projects**, następnie wybór z kolejnej listy pozycji **Other**

The screenshot shows an IDE interface with a context menu open over a project named 'Wypożyczalnia' in the 'Files' view. The menu is divided into two columns. The left column contains standard project actions like 'New', 'Build', 'Clean and Build', 'Clean', 'Verify', 'Generate Javadoc', 'Run', 'Deploy', 'Debug', 'Profile', 'Test RESTful Web Services', 'Test', 'Open Required Projects', 'Close', 'Rename...', 'Move...', 'Copy...', 'Delete', 'Find...', 'Inspect and Transform...', and 'Versioning'. The right column contains options for creating new artifacts: 'Validation Constraint...', 'JSF Managed Bean...', 'JSF Page...', 'Java Class...', 'Java Package...', 'JavaBeans Component...', 'JFrame Form...', 'Property Editor...', 'Frame Form...', 'JSF Pages from Entity Classes...', 'Entity Classes from Database...', 'Facelets Template Client...', 'Facelets Template...', 'JSF Page...', and 'TestNG Test Case...'. The 'Other...' option at the bottom of the right column is highlighted. In the background, the 'index.xhtml' file is open, showing XML code with a JNDI reference: `<resource-ref name="jdbc/Katalogksiazek" type="javax.sql.DataSource" url="java:comp/env/jdbc/Katalogksiazek"/>`. The 'Palette' on the right shows various UI components like 'Table', 'Form', 'Text Input', etc.

Wykonanie odniesienia typu JNDI do bazy danych Katalogsiazek (2): wybór **GlassFish/JDBC Resource, Next**



Wykonanie odniesienia typu JNDI do bazy danych Katalogksiazek (3): wybór **Create New JDBC Connection Pool** oraz wpisanie nazwy **JNDI Name: jdbc/wypożyczalnia, Next**

New JDBC Resource

Steps

1. Choose ...
- 2. General Attributes - JDBC Resource**
3. Properties
4. Choose Database Connection
5. Add Connection Pool Properties
6. Add Connection Pool Optional Properties

General Attributes

Provide configuration information for the JDBC Resource.
Either choose an existing JDBC Connection Pool, or create a new JDBC Connection Pool.
Fields with an * mark are required.

Use Existing JDBC Connection Pool

< No JDBC Connection Pool >

Create New JDBC Connection Pool

JNDI Name:*

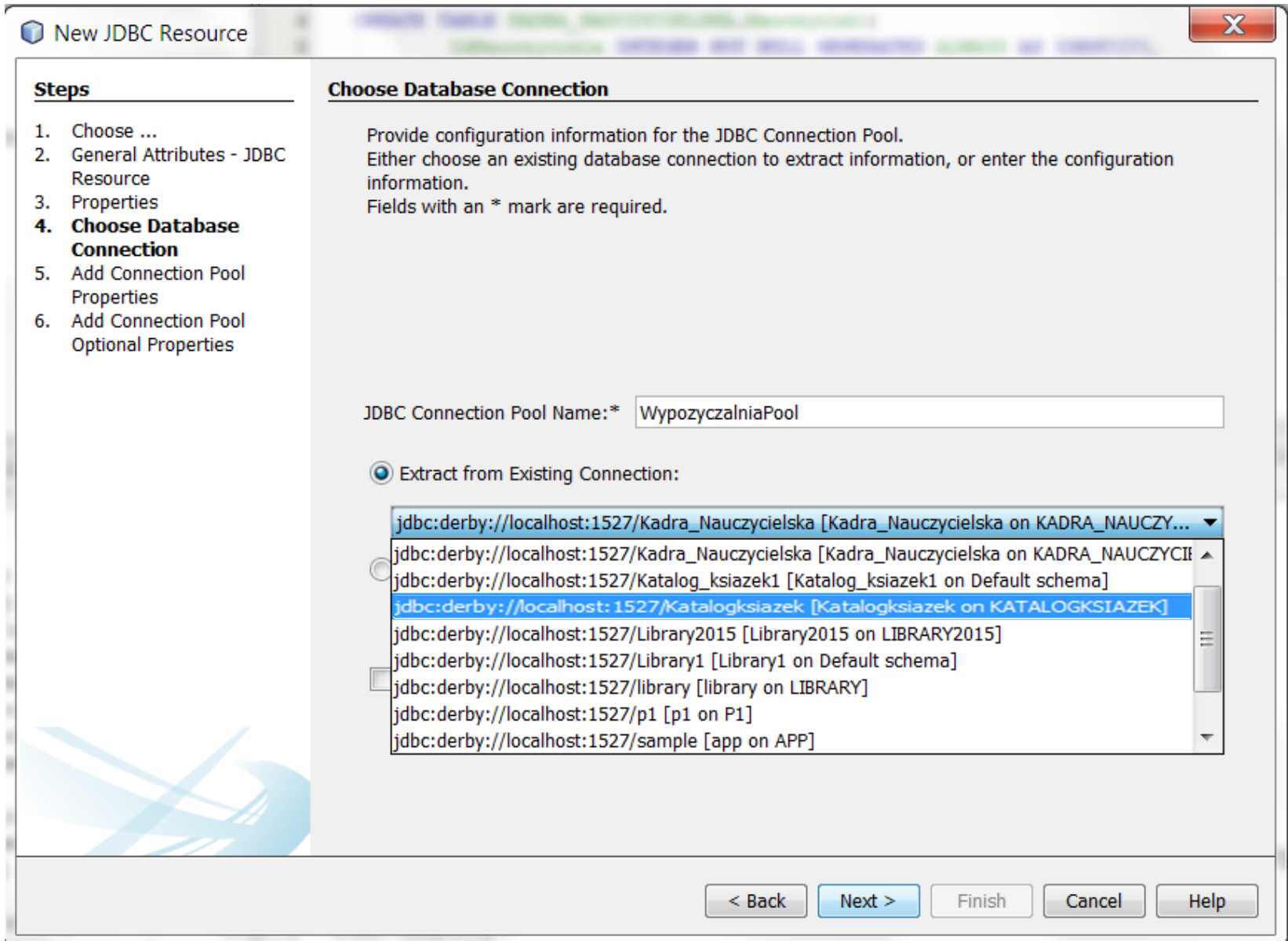
Object Type:

Enabled:

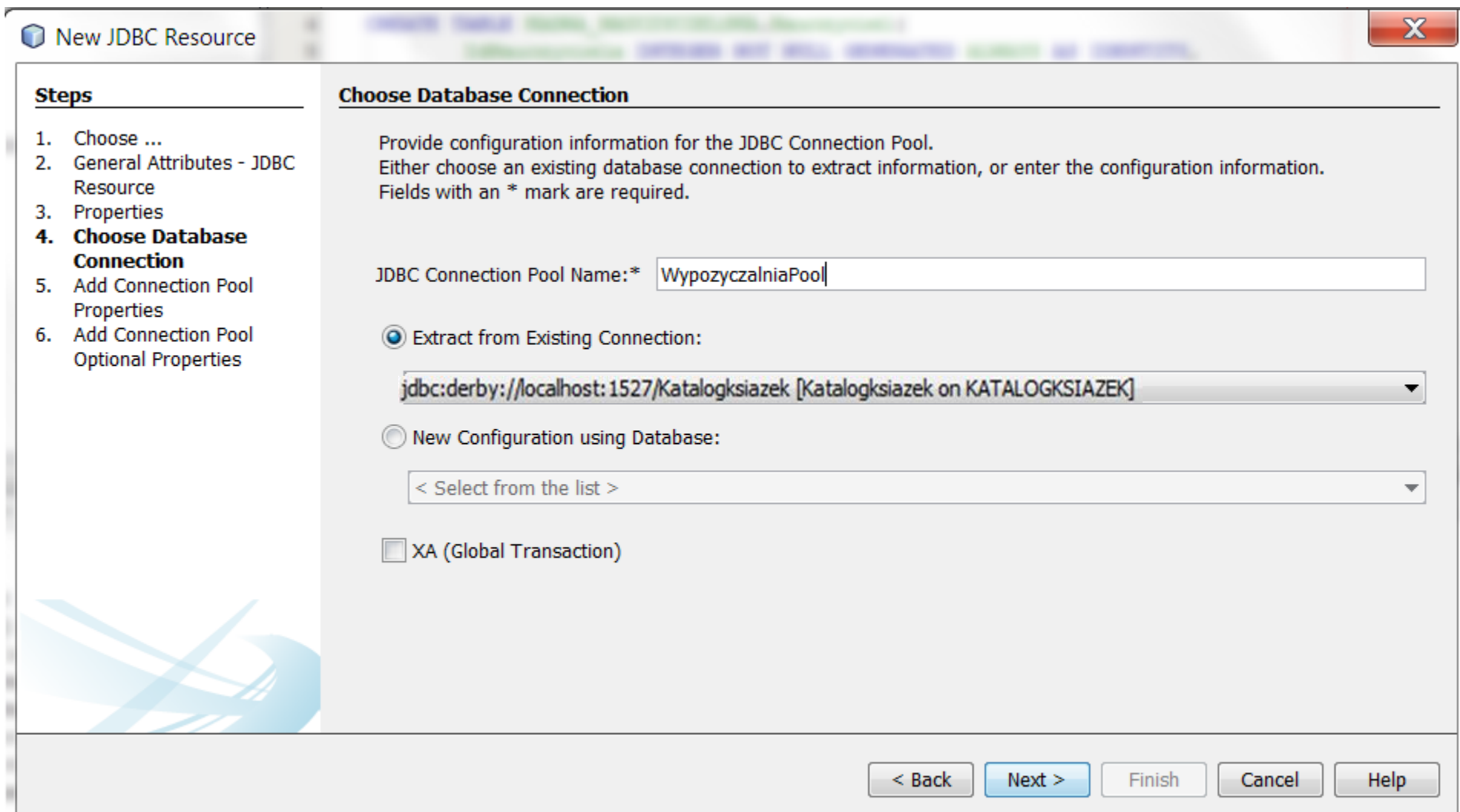
Description:

< Back **Next >** Finish Cancel Help

Wykonanie odniesienia typu JNDI do bazy danych Katalogksiazek (4): wpisanie nazwy WypozyczalniaPool w polu **JDBC Connection Pool Name** oraz wybór z listy **Extract from Existing Connection** połączenia do bazy danych, **Next**



Wykonanie odniesienia typu JNDI do bazy danych Katalogksiazek (5): wpisanie nazwy WypozyczalniaPool w polu **JDBC Connection Pool Name** oraz wybór z listy **Extract from Existing Connection** połączenia do bazy danych - rezultat , **Next**



The screenshot shows a window titled "New JDBC Resource" with a close button in the top right corner. On the left, a "Steps" sidebar lists the following steps:

1. Choose ...
2. General Attributes - JDBC Resource
3. Properties
- 4. Choose Database Connection**
5. Add Connection Pool Properties
6. Add Connection Pool Optional Properties

The main area is titled "Choose Database Connection" and contains the following text:

Provide configuration information for the JDBC Connection Pool.
Either choose an existing database connection to extract information, or enter the configuration information.
Fields with an * mark are required.

The "JDBC Connection Pool Name:" field contains the text "WypozyczalniaPool".

There are two radio button options:

- Extract from Existing Connection:
A dropdown menu below this option shows "jdbc:derby://localhost:1527/Katalogksiazek [Katalogksiazek on KATALOGKSIAZEK]".
- New Configuration using Database:
A dropdown menu below this option shows "< Select from the list >".

At the bottom, there is a checkbox labeled "XA (Global Transaction)" which is currently unchecked.

At the bottom right, there are five buttons: "< Back", "Next >", "Finish", "Cancel", and "Help". The "Next >" button is highlighted in blue.

Wykonanie odniesienia typu JNDI do bazy danych Katalogksiazek (6): Next

New JDBC Resource

Steps

1. Choose ...
2. General Attributes - JDBC Resource
3. Properties
4. Choose Database Connection
- 5. Add Connection Pool Properties**
6. Add Connection Pool Optional Properties

Add Connection Pool Properties

Enter the Datasource Classname, URL, and User to continue.
Hit the Enter key to save values in the Properties table.

Datasource Classname:

Resource Type:

Description:

Properties:

Name	Value
URL	jdbc:derby://localhost:1527/Katalog_ksiazek
serverName	localhost
PortNumber	1527
DatabaseName	Katalog_ksiazek

Wykonanie odniesienia typu JNDI do bazy danych Katalogksiazek (7): Finish

New JDBC Resource

Steps

1. Choose ...
2. General Attributes - JDBC Resource
3. Properties
4. Choose Database Connection
5. Add Connection Pool Properties
- 6. Add Connection Pool Optional Properties**

Specify Optional Properties for Connection Pool

Pool Settings

Steady Pool Size:

Max Pool Size:

Max Wait Time:

Pool Resize Quantity:

Idle Timeout (secs):

Transaction Isolation

Transaction Isolation:

Guarantee Isolation Level:

Connection Validation

Connection Validation Required:

Validation Method:

Table Name:

Fail All Connections:

Non Transactional Connections:

Allow Non Component Callers:

< Back Next > **Finish** Cancel Help

Wykonanie klas typu **Entity** z bazy danych Katalogksiazek (1) - wybór **New** z listy po kliknięciu prawym klawiszem na nazwę projektu w zakładce **Projects**, następnie wybór z kolejnej listy pozycji **Other**

The screenshot shows an IDE interface with a context menu open over a project named 'Wypożyczalnia' in the 'Projects' view. The menu is divided into two panes. The left pane contains standard IDE actions such as 'New', 'Build', 'Clean and Build', 'Clean', 'Verify', 'Generate Javadoc', 'Run', 'Deploy', 'Debug', 'Profile', 'Test RESTful Web Services', 'Test', 'Open Required Projects', 'Close', 'Rename...', 'Move...', 'Copy...', 'Delete', 'Find...', 'Inspect and Transform...', and 'Versioning'. The right pane lists various code generation options, including 'Validation Constraint...', 'JSF Managed Bean...', 'JSF Page...', 'Java Class...', 'Java Package...', 'JavaBeans Component...', 'JFrame Form...', 'Property Editor...', 'Frame Form...', 'JSF Pages from Entity Classes...', 'Entity Classes from Database...', 'Facelets Template Client...', 'Facelets Template...', 'JSF Page...', and 'TestNG Test Case...'. The 'Other...' option at the bottom of the right pane is highlighted. In the background, the IDE's main editor displays an XML document with the following content:

```
<?xml version="1.0" encoding="UTF-8" ?>
<C//DTD XHTML 1.0
rg/1999/xhtml"
un.com/jsf/html"
</title>
```

The IDE's interface includes a menu bar (File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help), a toolbar, and a 'Palette' on the right side with categories like 'HTML', 'HTML Forms', and 'JSF'. The status bar at the bottom indicates '4 | 8 | 13 | INS'.

Wykonanie klas typu **Entity** z bazy danych Katalogksiazek (2) – wybór **Persistence/Entity Classes from Database**

New File

Steps

1. **Choose File Type**
2. ...

Choose File Type

Project:

Categories:

- Contexts and Dependency
- Java
- Swing GUI Forms
- JavaBeans Objects
- AWT GUI Forms
- Unit Tests
- Selenium
- Persistence**
- Groovy
- Hibernate

File Types:

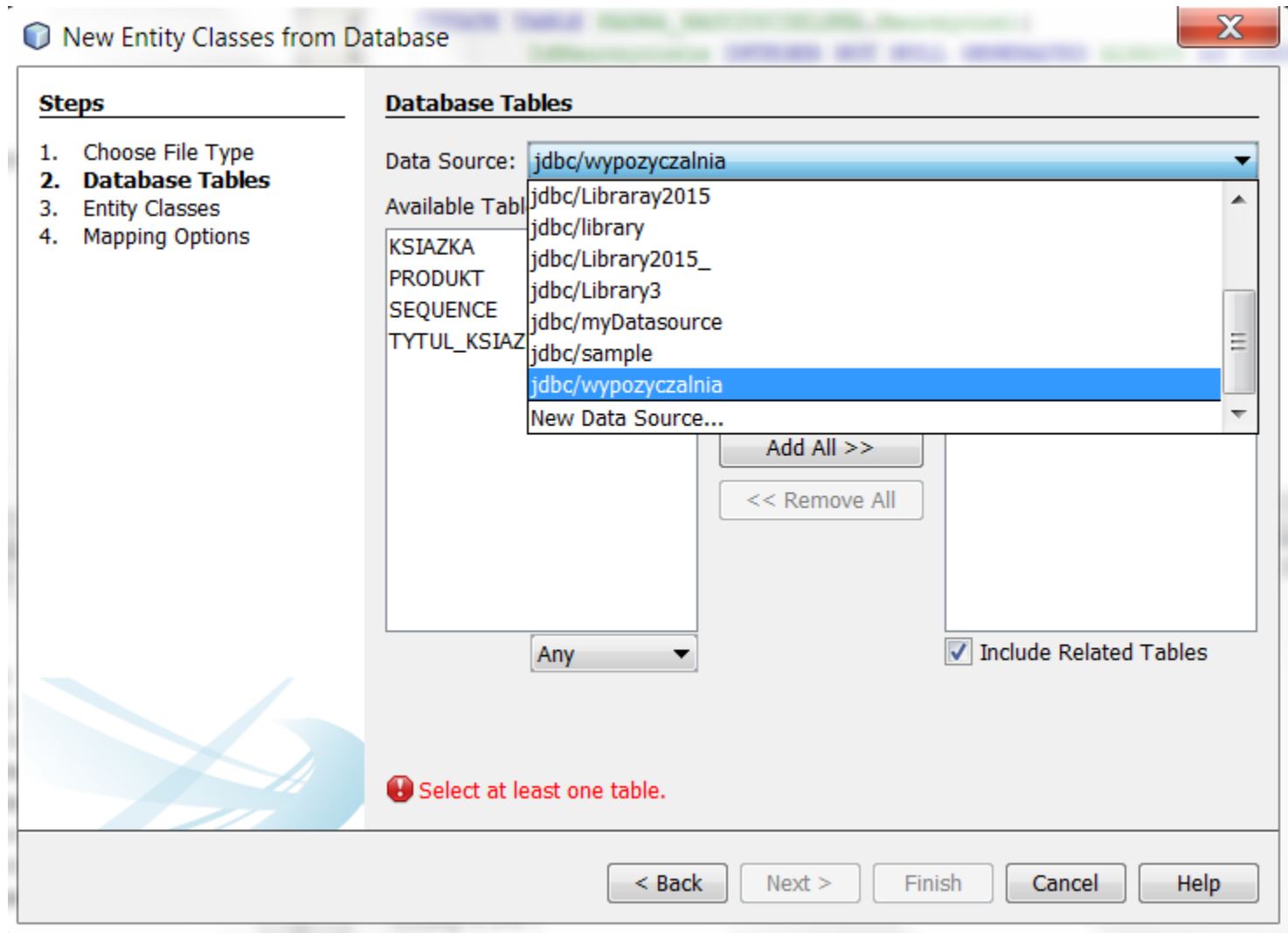
- Entity Class
- Entity Classes from Database**
- JPA Controller Classes from Entity Classes
- JSF Pages from Entity Classes
- Persistence Unit
- Database Schema
- Session Beans For Entity Classes
- RESTful Web Services from Entity Classes
- RESTful Web Services from Database

Description:

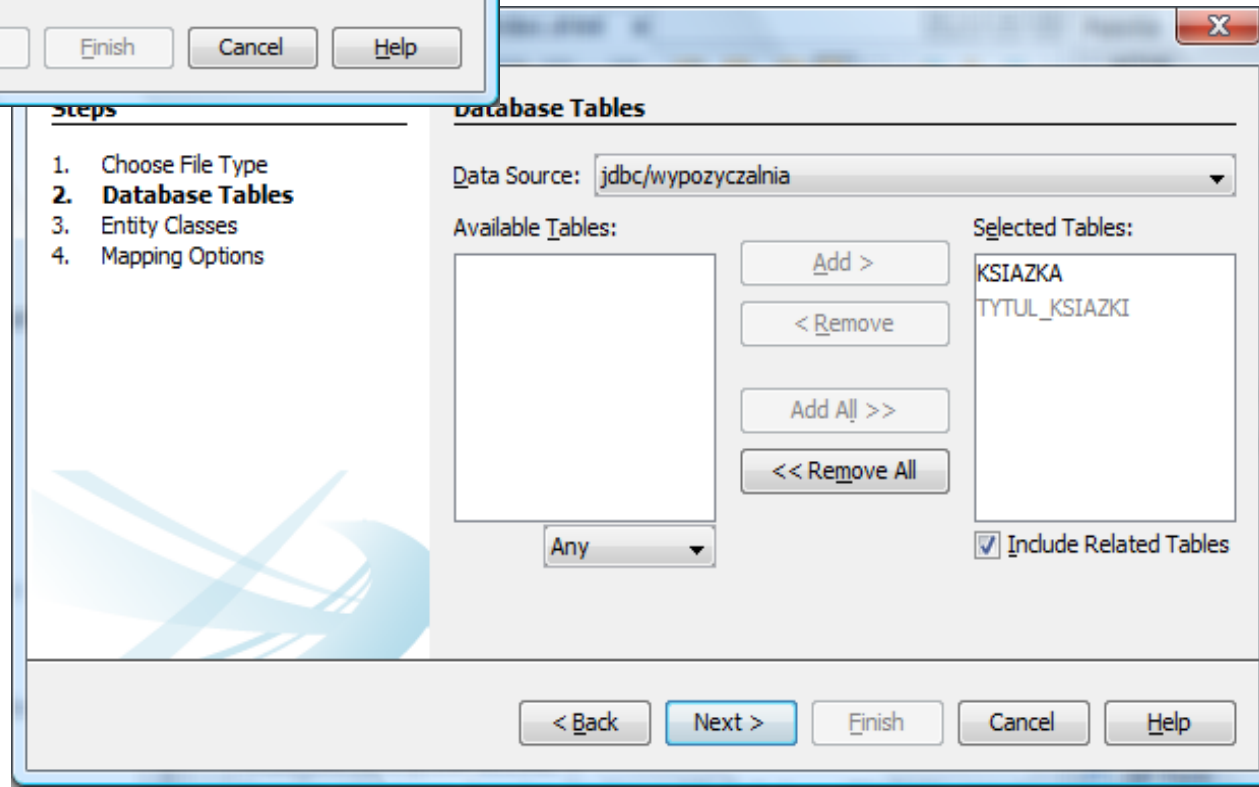
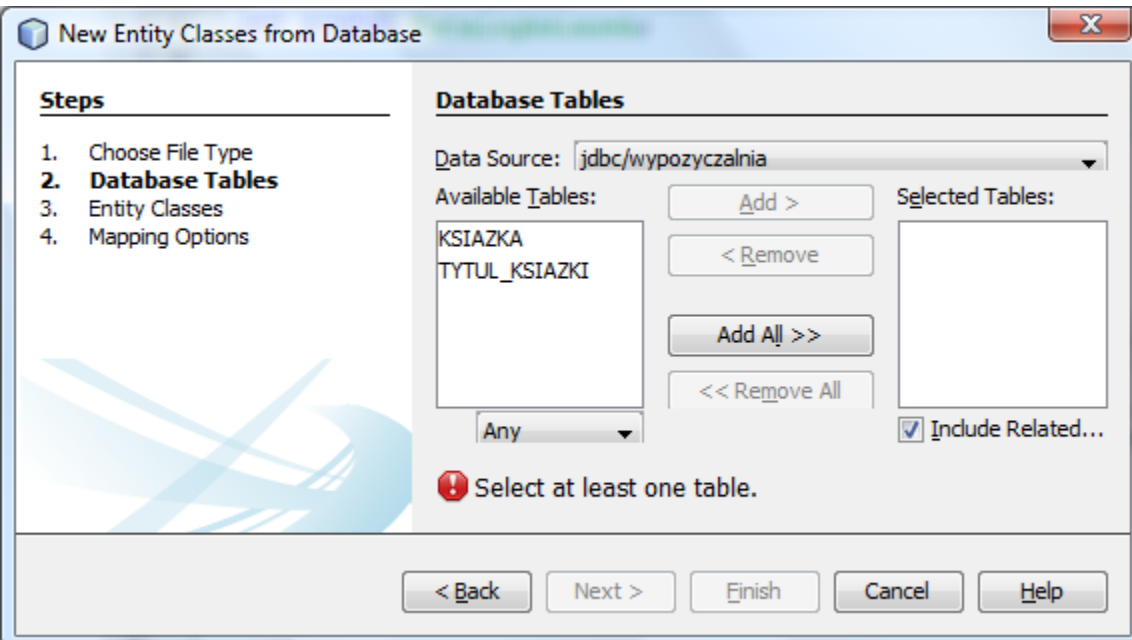
Creates Java Persistence API entity classes based on an existing relational database. Entity classes are used to represent objects whose lifespan is longer than a typical program execution. This template creates an entity class for each selected table, complete with named query annotations, fields representing columns, and relationships representing foreign keys.

< Back **Next >** Finish Cancel Help

Wykonanie klas typu **Entity** z bazy danych (3)– wybór odniesienia typu JNDI do bazy danych Katalogksiazek: **jdbc/wypożyczalnia**



Wybór tabel do mapowania - wybór **Add All**



Wykonanie klas typu Entity z bazy danych (5)

New Entity Classes from Database

Steps

1. Choose File Type
2. Database Tables
3. **Entity Classes**
4. Mapping Options

Entity Classes

Specify the names and the location of the entity classes.

Class Names:

Database Table	Class Name	Generation Type
KSIAZKA	Ksiazka	New
TYTUL_KSIAZKI	TytulKsiazki	New

Project: Wypożyczalnia

Location: Source Packages

Package: jpa

Generate Named Query Annotations for Persistent Fields

Generate JAXB Annotations

Create Persistence Unit

< Back Next > Finish Cancel Help

Wygenerowana klasa typu **Entity** Ksiazka

```
package jpa;

import java.io.Serializable;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;
import javax.validation.constraints.NotNull;
import javax.xml.bind.annotation.XmlRootElement;
```

@Entity

@Table(name = "KSIAZKA")

@XmlRootElement

@NamedQueries({

 @NamedQuery(name = "Ksiazka.findAll", query = "SELECT k FROM Ksiazka k"),

 @NamedQuery(name = "Ksiazka.findByKsiazkald", query = "SELECT k FROM Ksiazka k
 WHERE k.ksiazkald = :ksiazkald"),

 @NamedQuery(name = "Ksiazka.findByNumer", query = "SELECT k FROM Ksiazka k
 WHERE k.numer = :numer"))})

public class Ksiazka **implements** Serializable {

private static final long serialVersionUID = 1L;

 @Id

 @GeneratedValue(strategy = GenerationType.IDENTITY)

 @Basic(optional = false)

 @Column(name = "KSIAZKA_ID")

private Integer ksiazkald;

 @Basic(optional = false)

 @NotNull

 @Column(name = "NUMER")

private int numer;

 @JoinColumn(name = "ID_TYTUL", referencedColumnName = "TYTUL_ID")

 @ManyToOne(optional = false)

private TytulKsiazki idTytul;


```
public Ksiazka() { }
public Ksiazka(Integer ksiazkald) {
    this.ksiazkald = ksiazkald;
}
public Ksiazka(Integer ksiazkald, int numer) {
    this.ksiazkald = ksiazkald;
    this.numer = numer;
}
public Integer getKsiazkald() {
    return ksiazkald;
}
public void setKsiazkald(Integer ksiazkald) {
    this.ksiazkald = ksiazkald;
}
public int getNumer() {
    return numer;
}
public void setNumer(int numer) {
    this.numer = numer;
}
public TytulKsiazki getIdTytul() {
    return idTytul;
}
public void setIdTytul(TytulKsiazki idTytul) {
    this.idTytul = idTytul;
}
```

@Override

```
public int hashCode() {  
    int hash = 0;  
    hash += (ksiazkald != null ? ksiazkald.hashCode() : 0);  
    return hash;  
}
```

@Override

```
public boolean equals(Object object) {  
    // TODO: Warning - this method won't work in the case the id fields are not set  
    if (!(object instanceof Ksiazka)) {  
        return false;  
    }  
    Ksiazka other = (Ksiazka) object;  
    if ((this.ksiazkald == null && other.ksiazkald != null) || (this.ksiazkald != null &&  
        !this.ksiazkald.equals(other.ksiazkald))) {  
        return false;  
    }  
    return true;  
}
```

@Override

```
public String toString() {  
    return "jpa.Ksiazka[ ksiazkald=" + ksiazkald + " ]";  
}
```

}

Wygenerowana klasa typu **Entity**Tytul_ksiazki

```
package jpa;
import java.io.Serializable;
import java.util.Collection;
import javax.persistence.Basic;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.OneToOne;
import javax.persistence.Table;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;
import javax.xml.bind.annotation.XmlRootElement;
import javax.xml.bind.annotation.XmlTransient;
```

@Entity

@Table(name = "TYTUL_KSIAZKI")

@XmlRootElement

@NamedQueries({

 @NamedQuery(name = "TytulKsiazki.findAll", query = "SELECT t FROM TytulKsiazki t"),

 @NamedQuery(name = "TytulKsiazki.findByTytulId", query = "SELECT t FROM TytulKsiazki t WHERE t.tytulId = :tytulId"),

 @NamedQuery(name = "TytulKsiazki.findByTytul", query = "SELECT t FROM TytulKsiazki t WHERE t.tytul = :tytul"),

 @NamedQuery(name = "TytulKsiazki.findByAutorNazwisko", query = "SELECT t FROM TytulKsiazki t WHERE t.autorNazwisko = :autorNazwisko"),

 @NamedQuery(name = "TytulKsiazki.findByAutorImie", query = "SELECT t FROM TytulKsiazki t WHERE t.autorImie = :autorImie"),

 @NamedQuery(name = "TytulKsiazki.findByIsbn", query = "SELECT t FROM TytulKsiazki t WHERE t.isbn = :isbn"),

 @NamedQuery(name = "TytulKsiazki.findByWydawnictwo", query = "SELECT t FROM TytulKsiazki t WHERE t.wydawnictwo = :wydawnictwo"))})

```
public class TytulKsiazki implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(name = "TYTUL_ID")
    private Integer tytulId;
    @Basic(optional = false)
    @NotNull
    @Size(min = 1, max = 50)
    @Column(name = "TYTUL")
    private String tytul;
    @Basic(optional = false)
    @NotNull
    @Size(min = 1, max = 50)
    @Column(name = "AUTOR_NAZWISKO")
    private String autorNazwisko;
    @Basic(optional = false)
    @NotNull
    @Size(min = 1, max = 50)
    @Column(name = "AUTOR_IMIE")
    private String autorImie;
    @Basic(optional = false)
    @NotNull
    @Size(min = 1, max = 50)
    @Column(name = "ISBN")
    private String isbn;
```

```
@Basic(optional = false)
@NotNull
@Size(min = 1, max = 50)
@Column(name = "WYDAWNICTWO")
private String wydawnictwo;
@OneToMany(cascade = CascadeType.ALL, mappedBy = "idTytul")
private Collection<Ksiazka> ksiazkaCollection
```

```
public TytulKsiazki() {
}
public TytulKsiazki(Integer tytulId) {
    this.tytulId = tytulId;
}
public TytulKsiazki(Integer tytulId, String tytul, String autorNazwisko, String autorImie, String isbn,
String wydawnictwo) {
    this.tytulId = tytulId;
    this.tytul = tytul;
    this.autorNazwisko = autorNazwisko;
    this.autorImie = autorImie;
    this.isbn = isbn;
    this.wydawnictwo = wydawnictwo;
}
public String getTytul() {
    return tytul;
}
public void setTytul(String tytul) {
    this.tytul = tytul;
}
```

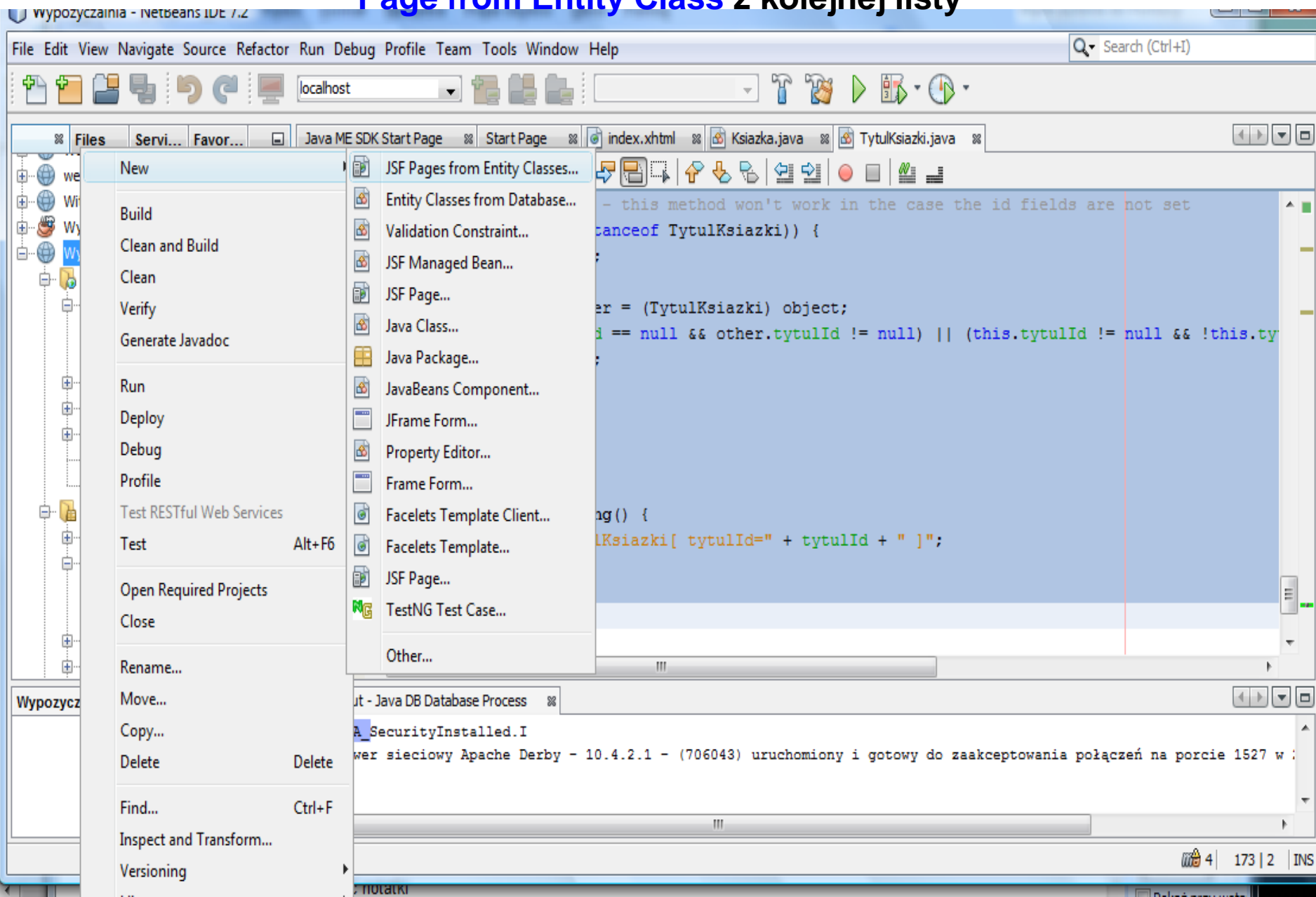
```
public String getAutorNazwisko() {
    return autorNazwisko;
}
public void setAutorNazwisko(String autorNazwisko) {
    this.autorNazwisko = autorNazwisko;
}
public String getAutorImie() {
    return autorImie;
}
public void setAutorImie(String autorImie) {
    this.autorImie = autorImie;
}
public String getIsbn() {
    return isbn;
}
public void setIsbn(String isbn) {
    this.isbn = isbn;
}
public String getWydawnictwo() {
    return wydawnictwo;
}
public void setWydawnictwo(String wydawnictwo) {
    this.wydawnictwo = wydawnictwo;
}
public Integer getTytulId() {
    return tytulId;
}
public void setTytulId(Integer tytulId) {
    this.tytulId = tytulId;
}
}
```

```

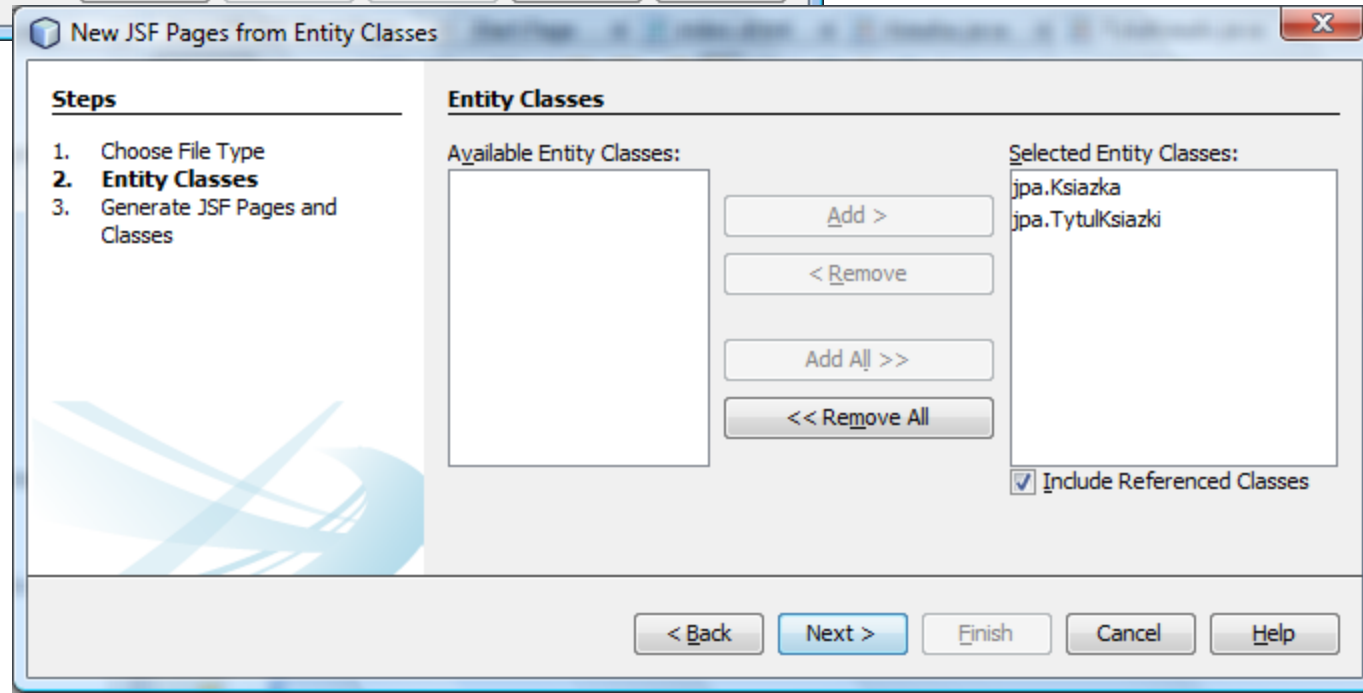
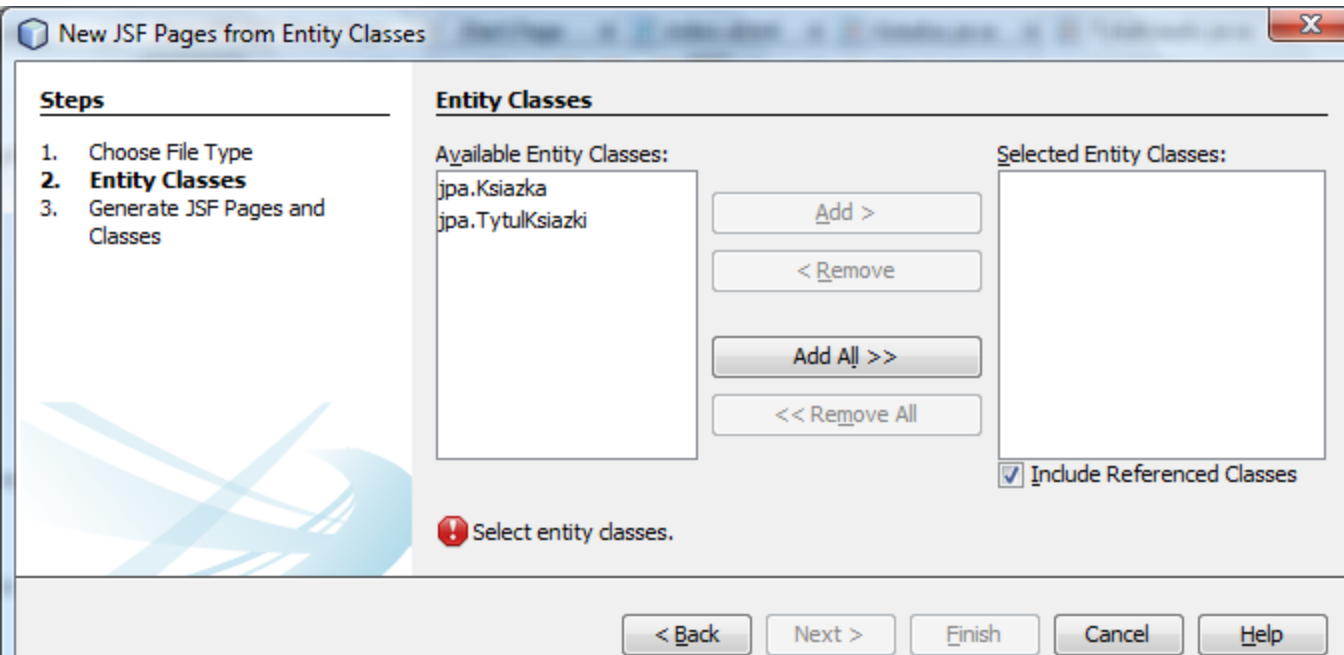
@XmlTransient
public Collection<Ksiazka> getKsiazkaCollection() {
    return ksiazkaCollection;
}
public void setKsiazkaCollection(Collection<Ksiazka> ksiazkaCollection) {
    this.ksiazkaCollection = ksiazkaCollection;
}
@Override
public int hashCode() {
    int hash = 0;
    hash += (tytulId != null ? tytulId.hashCode() : 0);
    return hash;
}
@Override
public boolean equals(Object object) {
    // TODO: Warning - this method won't work in the case the id fields are not set
    if (!(object instanceof TytulKsiazki)) {
        return false;
    }
    TytulKsiazki other = (TytulKsiazki) object;
    if ((this.tytulId == null && other.tytulId != null) || (this.tytulId != null &&
!this.tytulId.equals(other.tytulId))) {
        return false;
    }
    return true;
}
@Override
public String toString() {
    return "jpa.TytulKsiazki[ tytulId=" + tytulId + " ]";
}
}

```


Generowanie stron JSF z encji (1) – wybór **New** z listy po kliknięciu prawym klawiszem na nazwę projektu w zakładce **Projects**, następnie wybór pozycji **JSP Page from Entity Class** z kolejnej listy



Generowanie stron JSF z encji (2)



Generowanie stron JSF z encji (3)

New JSF Pages from Entity Classes

Steps

1. Choose File Type
2. Entity Classes
3. **Generate JSF Pages and Classes**

Generate JSF Pages and Classes

Specify the package of existing or new EJBs and the package of JSF classes.

Project:

Location:

Session Bean Package:

JSF Classes Package:

Specify the location of new JSF pages.

JSF Pages Folder:

Localization Bundle Name:

Override existing files

Customize Template:

< Back Next > **Finish** Cancel Help

Zawartość katalogu projektu aplikacji internetowej (1)

The screenshot displays the NetBeans IDE 7.2 interface. On the left, the 'Files' view shows a project named 'Wypożyczalnia' with a 'Web Pages' folder containing several subfolders and files. A red box highlights the 'Web Pages' folder and its contents. The main editor window shows the source code of 'index.xhtml', which is an XHTML page with a title 'Facelet Title' and two links. A red arrow points from the 'index.xhtml' file in the 'Web Pages' folder to the code in the editor. A black box with white text is overlaid on the code, stating: 'Warstwa prezentacji: strony JSF do dynamicznego generowania stron www warstwy klienta'. The bottom status bar shows the output of a Java DB Database Process, indicating that the Apache Derby server is running.

```
html h:body
1 <?xml version='1.0' encoding='UTF-8' ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/I
3 <html xmlns="http://www.w3.org/1999/xhtml"
4     xmlns:h="http://java.sun.com/jsf/html">
5 <h:head>
6     <title>Facelet Title</title>
7     <h:outputStylesheet name="css/jsfcrud.css"/>
8 </h:head>
9 <h:body>
10     Hello from Facelets
11     <br />
12     <h:link outcome="/ksiazka/List" value="Show All Ksiazka Items"/>
13     <br />
14     <h:link outcome="/tytulKsiazki/List" value="Show All TytulKsiazki Items"/>
15 </h:body>
16
17 </html>
18
19
20
```

Warstwa prezentacji: strony JSF do dynamicznego generowania stron www warstwy klienta

Output - Java DB Database Process
DRDA_SecurityInstalled.I
Serwer sieciowy Apache Derby - 10.4.2.1 - (706043) uruchomiony i gotowy do zaakceptowania połączeń n

Zawartość katalogu projektu aplikacji internetowej (2)

Wypozyczalnia - NetBeans IDE 7.2

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

localhost

Pro... Files Services Favorit...

Wypozyczalnia

- Web Pages
- Source Packages
 - <default package>
 - Bundle.properties
 - jpa
 - Ksiazka.java
 - TytulKsiazki.java
 - jpa.session
 - AbstractFacade.java
 - KsiazkaFacade.java
 - TytulKsiazkiFacade.java
- jsf
 - KsiazkaController.java
 - TytulKsiazkiController.java
 - jsf.util
 - JsfUtil.java
 - PaginationHelper.java
- Libraries
 - JDK 1.7 (Default)
 - GlassFish Server 3+
 - Enterprise Beans
 - KsiazkaFacade
 - TytulKsiazkiFacade
 - Configuration Files
 - MANIFEST.MF
 - faces-config.xml
 - persistence.xml
 - web.xml
 - Server Resources
 - glassfish-resources.xml

Source History

```
1  /*...*/
5  package jpa.session;
6
7  import javax.ejb.Stateless;
8  import javax.persistence.EntityManager;
9  import javax.persistence.PersistenceContext;
10 import jpa.TytulKsiazki;
11
12 /*...*/
16 @Stateless
17 public class TytulKsiazkiFacade extends AbstractFacade<TytulKsiazki> {
18     @PersistenceContext(unitName = "WypozyczalniaPU")
19     private EntityManager em;
20
21     @Override
22     protected EntityManager getEntityManager() {
23         return em;
24     }
25
26     public TytulKsiazkiFacade() {
27         super(TytulKsiazki.class);
28     }
29 }
30
```

Output - Java DB Database Process

DRDA_SecurityInstalled.I

Serwer sieciowy Apache Derby - 10.4.2.1 - (706043) uruchomiony i gotowy do zaakceptowania

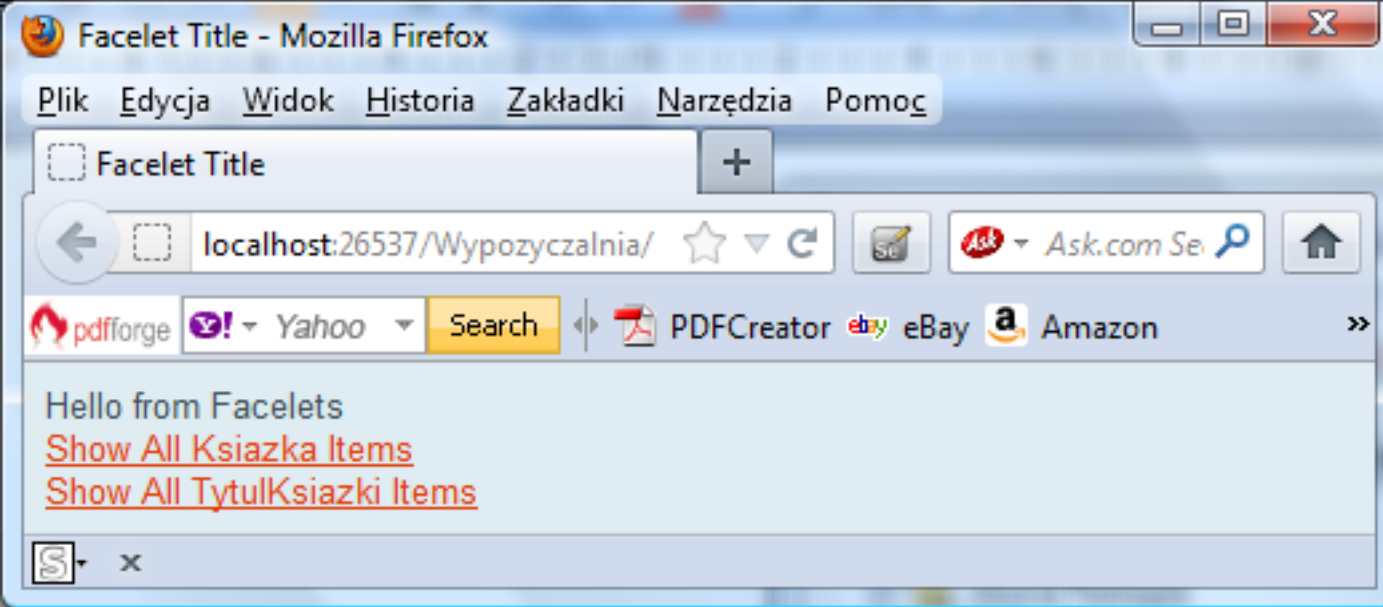
4 | 28 | 6 | INS

Warstwa prezentacji: strony JSF do dynamicznego generowania stron www warstwy klienta

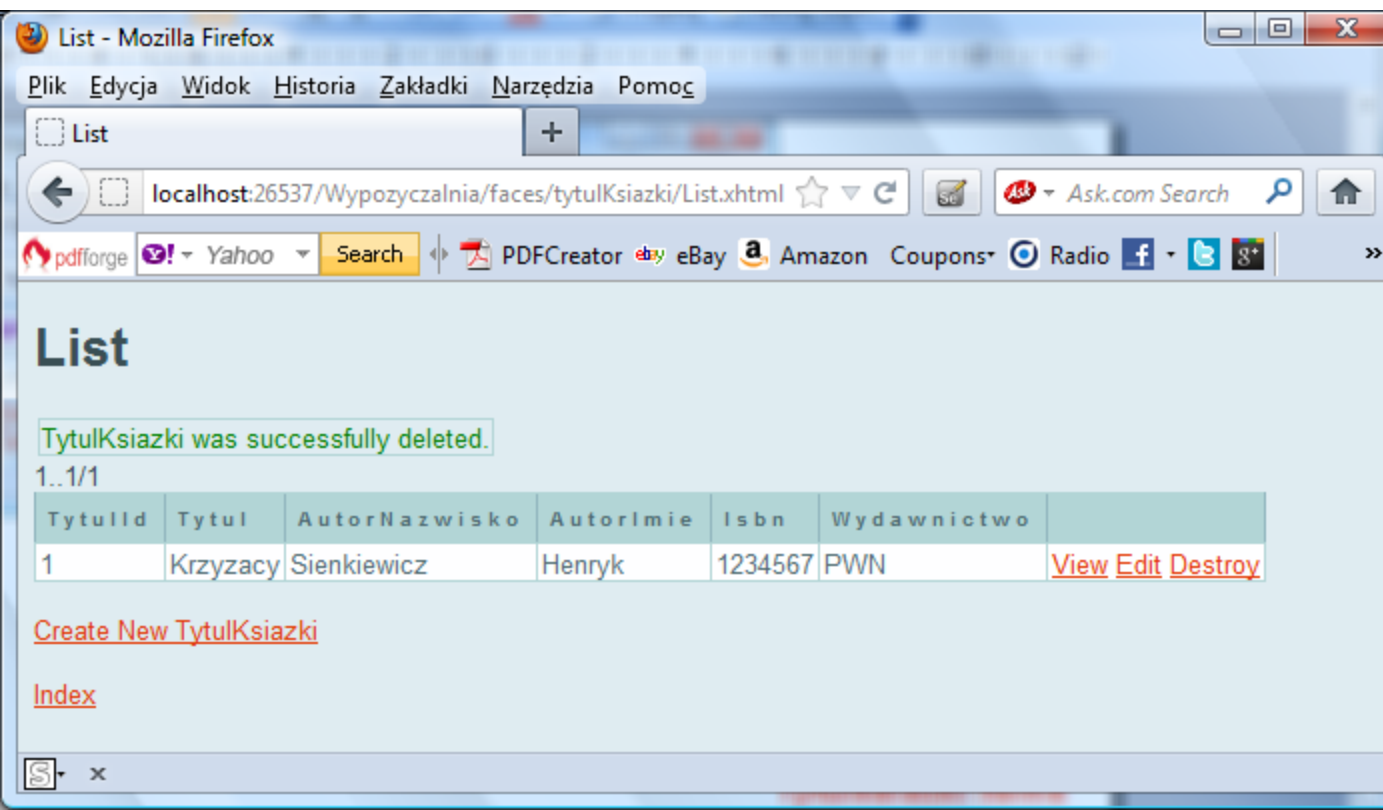
Warstwa integracji: obiektowy model danych

Warstwa biznesowa (ziarna EJB) + warstwa integracji (ORM)

Warstwa prezentacji

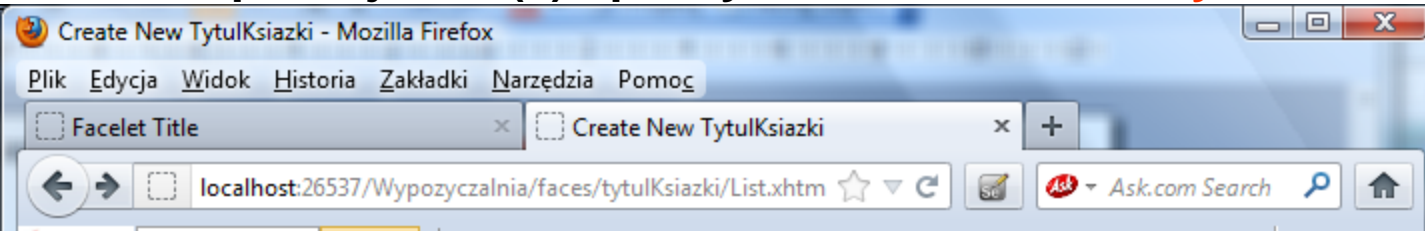


**Widok aplikacji
JSF (1) - po
uruchomieniu**

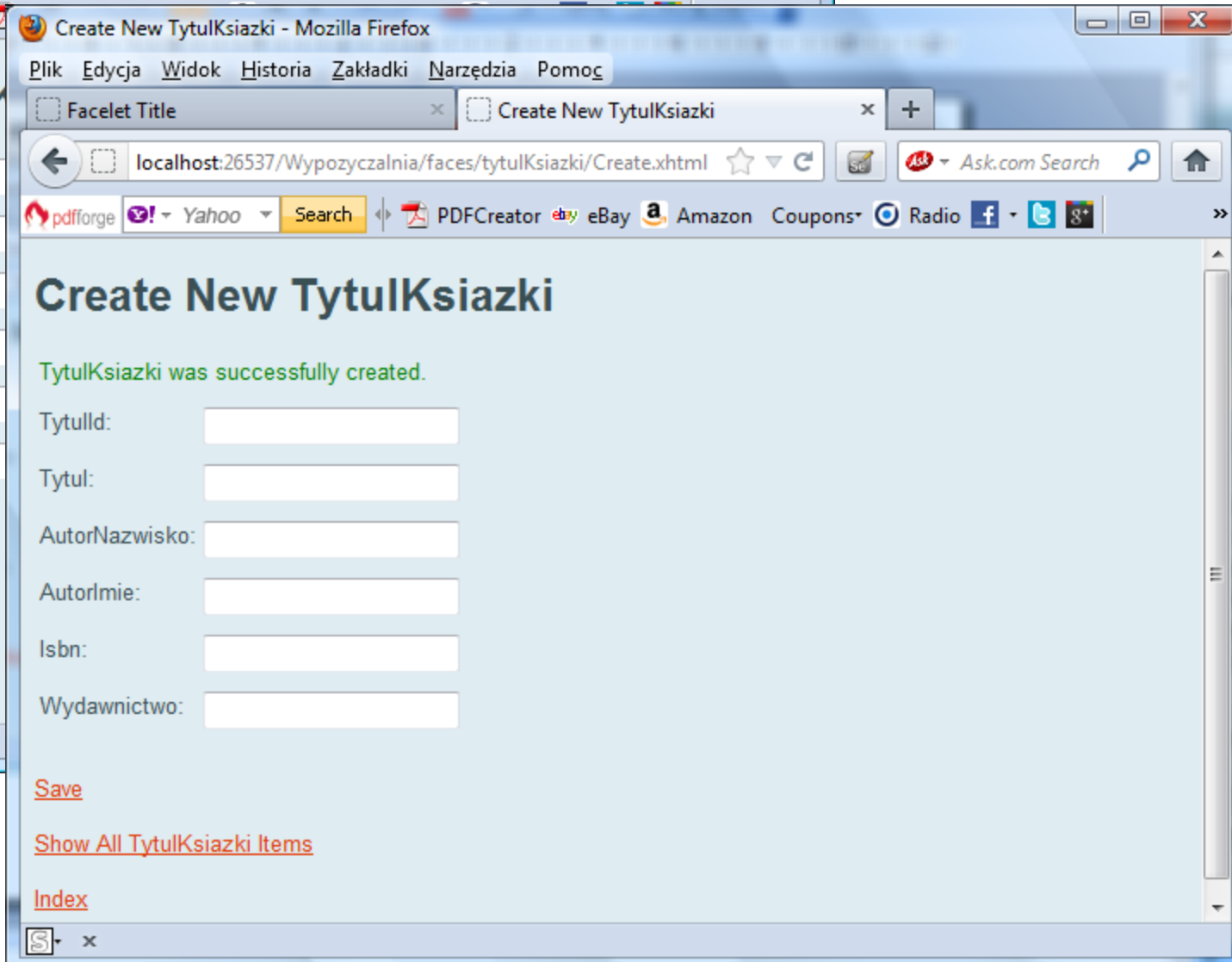
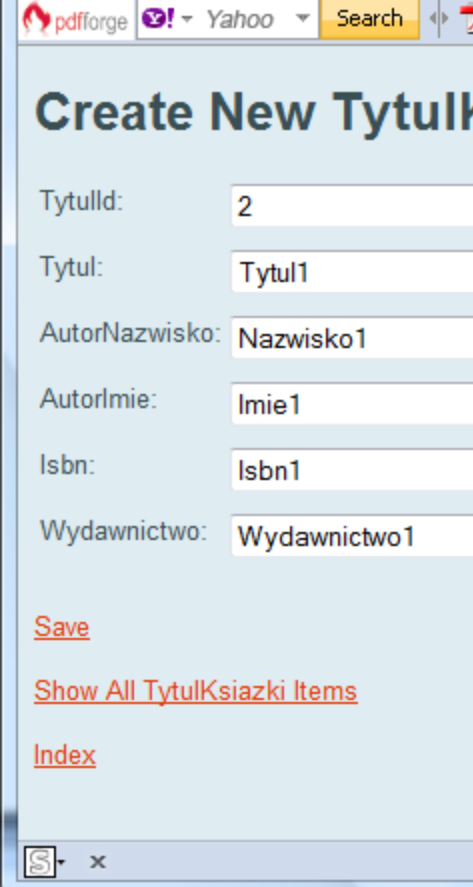


**Widok aplikacji
JSF (2) -
po wyborze
[Show All
TytułKsiążki Items](#)**

Widok aplikacji JSF (3) - po wyborze **Create New TytułKsiazki**



Widok aplikacji JSF
(4) - po wyborze
Save



Widok aplikacji JSF (5) - po wyborze **Show All TytulKsiazki Items**

The screenshot shows a Mozilla Firefox browser window with the following details:

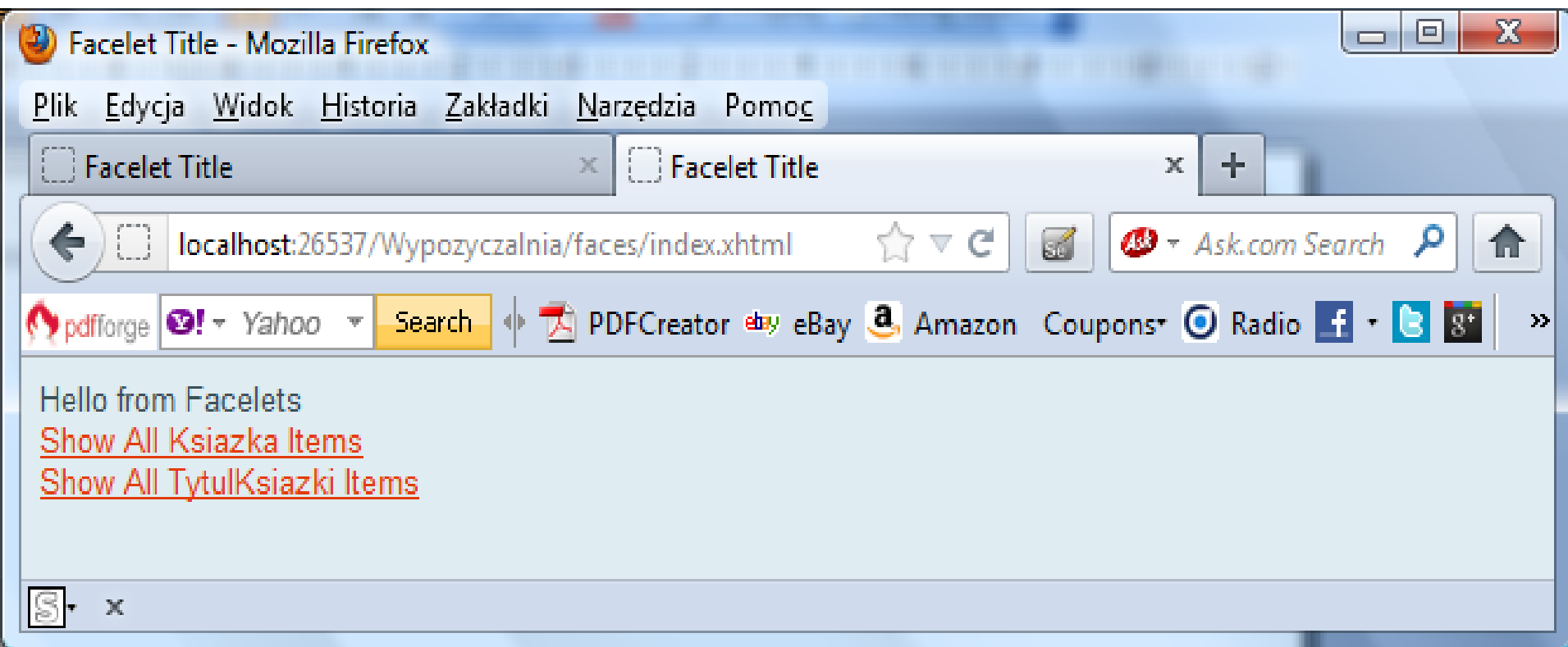
- Window title: List - Mozilla Firefox
- Address bar: localhost:26537/Wypożyczalnia/faces/tytulKsiazki/Create.xhtml
- Page title: List
- Page content: A table with 2 rows of book data and a link to 'Create New TytulKsiazki'.

TytuId	Tytuł	AutorNazwisko	AutorImie	Isbn	Wydawnictwo	
1	Krzyzacy	Sienkiewicz	Henryk	1234567	PWN	View Edit Destroy
3	Tytuł1	Nazwisko1	Imie1	Isbn1	Wydawnictwo1	View Edit Destroy

[Create New TytulKsiazki](#)

[Index](#)

Widok aplikacji JSF (6) - po wyborze **Index**



Widok aplikacji JSF (7) - po wyborze **Show All Ksiazka Items**

The screenshot shows a Mozilla Firefox browser window with the following elements:

- Window Title:** List - Mozilla Firefox
- Menu Bar:** Plik, Edycja, Widok, Historia, Zakładki, Narzędzia, Pomoc
- Tab Bar:** Two tabs are visible: 'Facelet Title' and 'List'.
- Address Bar:** localhost:26537/Wypożyczalnia/faces/ksiazka/List.xhtml
- Search Bar:** Ask.com Search
- Toolbar:** pdfforge, Yahoo Search, PDFCreator, eBay, Amazon, Coupons, Radio, Facebook, Twitter, Google+.
- Page Content:**
 - ## List
 - 1..1/1

KsiazkaId	Numer	IdTytuł	
1	1	1	View Edit Destroy

[Create New Ksiazka](#)

[Index](#)

Widok aplikacji JSF (8) po wyborze **Create New Ksiazka**

Create New Ksiazka - Mozilla Firefox

Plik Edycja Widok Historia Zakładki Narzędzia Pomoc

Create New Ksiazka

localhost:26537/Wypożyczalnia/faces/ksiazka/List.xhtml

Create New Ksiazka

Ksiazkald:

Numer:

IdTytul:

-
- jpa.TytulKsiazki[tytulId=1]
- jpa.TytulKsiazki[tytulId=3]

[Save](#)

[Show All Ksiazka Items](#)

[Index](#)

Widok aplikacji JSF (9) - po wyborze **Create New Ksiazka**

The screenshot shows a Mozilla Firefox browser window with the following details:

- Browser Title:** Create New Ksiazka - Mozilla Firefox
- Address Bar:** localhost:26537/Wypożyczalnia/faces/ksiazka/List.xhtml
- Page Title:** Create New Ksiazka
- Form Fields:**
 - KsiazkaId:** Input field containing the value "2".
 - Numer:** Input field containing the value "2".
 - IdTytul:** Dropdown menu with the selected value "jpa.TytulKsiazki[tytulId=1]".
- Navigation Links:** [Save](#), [Show All Ksiazka Items](#), and [Index](#).

Widok aplikacji JSF (10) - po wyborze **Save**

Create New Książka - Mozilla Firefox

Plik Edycja Widok Historia Zakładki Narzędzia Pomoc

Create New Książka +

localhost:26537/Wypożyczalnia/faces/ksiazka/Create.xhtml

pdfforge Yahoo Search PDFCreator eBay Amazon Coupons Radio f t g+

Create New Książka

Książka was successfully created.

Książkald:

Numer:

IdTytul:

[Save](#)

[Show All Książka Items](#)

Widok aplikacji JSF (11) - po wyborze **Show All Książka Items**

The screenshot shows a Mozilla Firefox browser window with the following details:

- Window Title: List - Mozilla Firefox
- Address Bar: localhost:26537/Wypożyczalnia/faces/ksiazka/Create.xhtml
- Page Title: List
- Page Content:
 - 1..2/2
 - Table with 4 columns: KsiążkaId, Numer, IdTytuł, and actions.
 - Links: [Create New Książka](#), [Index](#)

KsiążkaId	Numer	IdTytuł	
1	1	1	View Edit Destroy
2	2	1	View Edit Destroy

Podsumowanie

- Przedstawiono budowę wielowarstwowej aplikacji internetowej typu Enterprise – operacje na bazie danych oparto na technologii ORM. Podejście opierało się na tworzeniu aplikacji „dostępowej” do bazy danych
- Aplikację o takiej samej budowie można zbudować metodą klasycznej „inżynierii wprost”, szczególnie w przypadku **konieczności tworzenia usług w warstwie biznesowej** np. typu Data Mining:
 - po **wykonaniu obiektowego modelu danych w warstwie biznesowej** (metoda przedstawiona na wykładzie 1 i na wykładach z Podstaw Inżynierii Oprogramowania) i **po przekształceniu klas modelu na typ Entity za pomocą adnotacji**
 - można **dodać warstwę prezentacji i klienta**
 - oraz warstwę **integracji** o takiej samej strukturze jak w przedstawionym przykładzie. W takim przypadku należy jedynie utworzyć pustą bazę danych – **tabele zostaną wygenerowane po pierwszym odwołaniu do bazy danych**