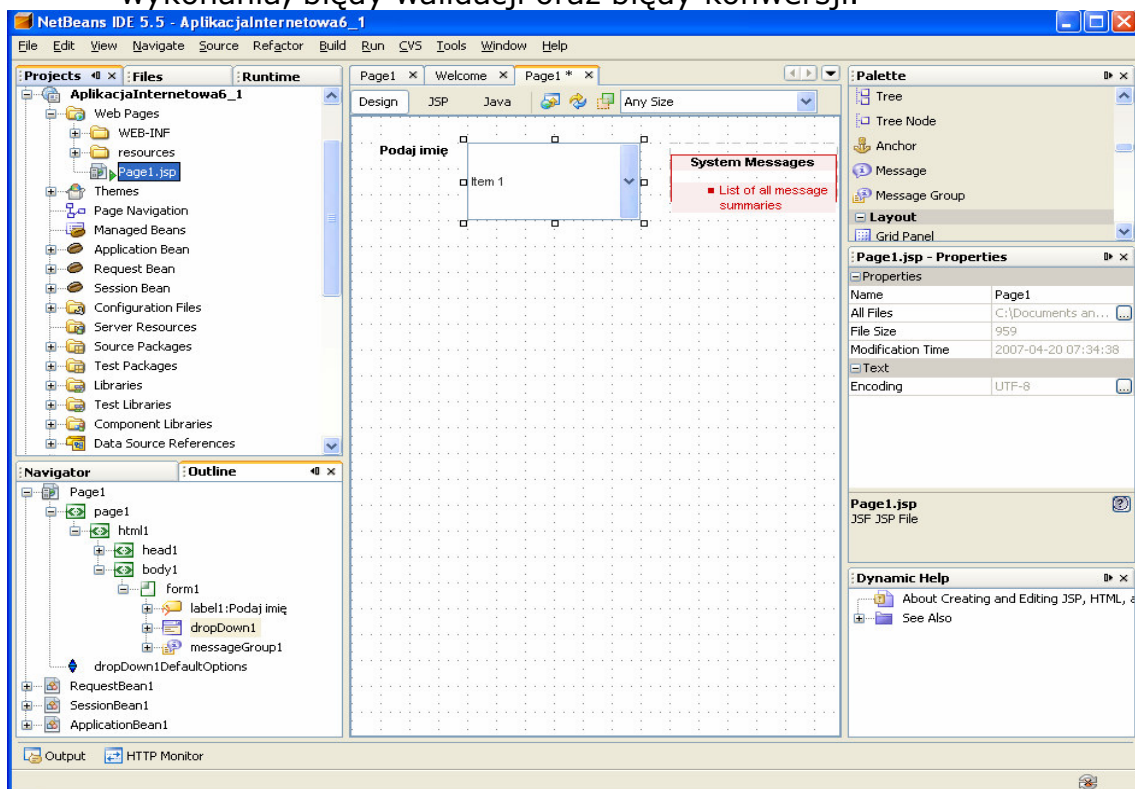


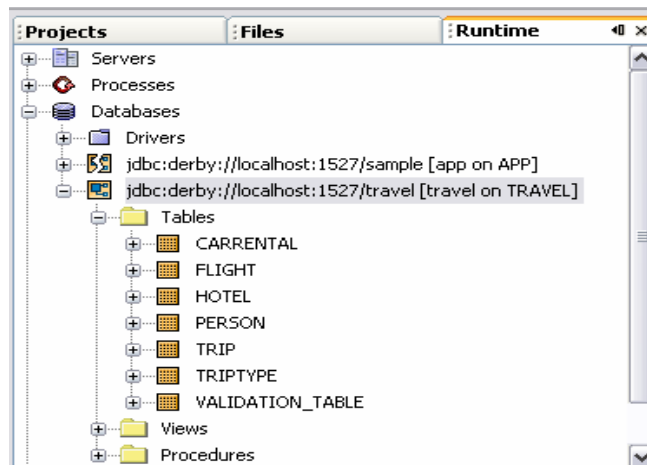
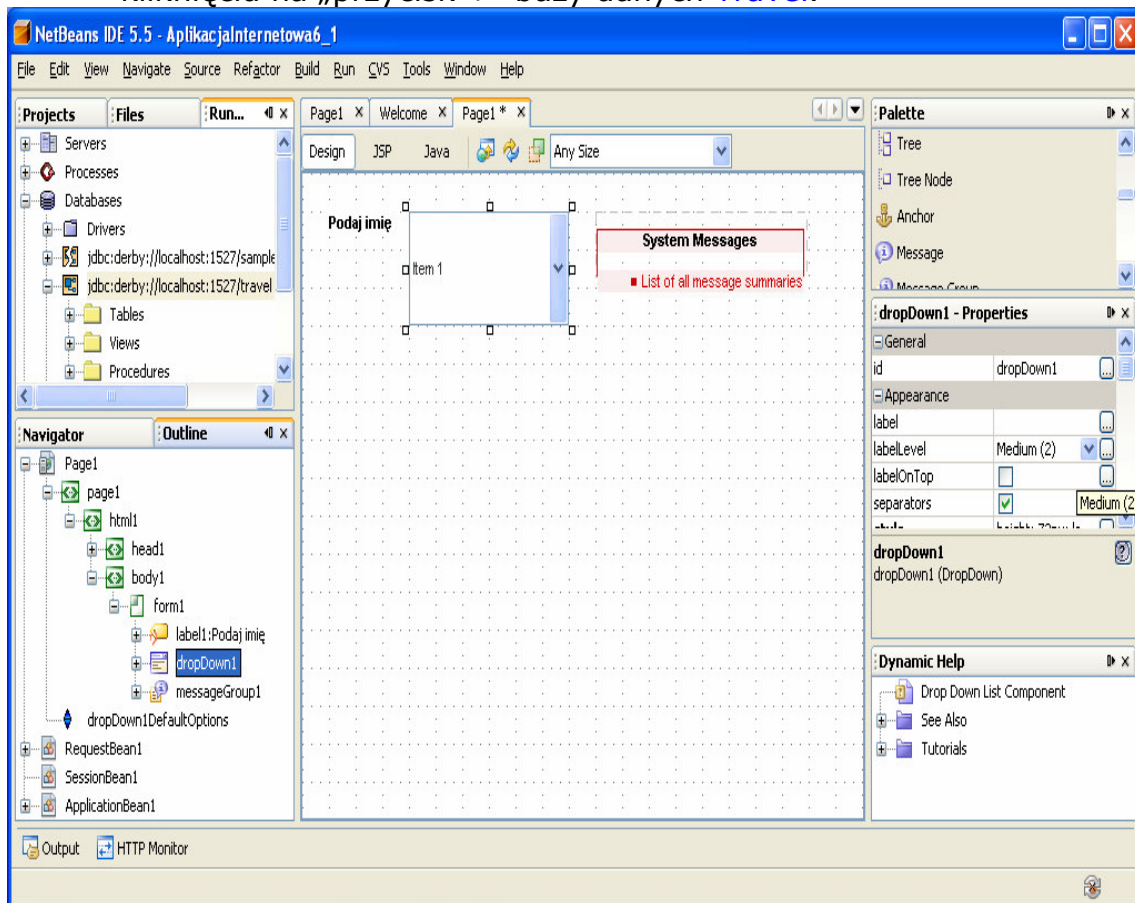
Lab3.

Zad.1.

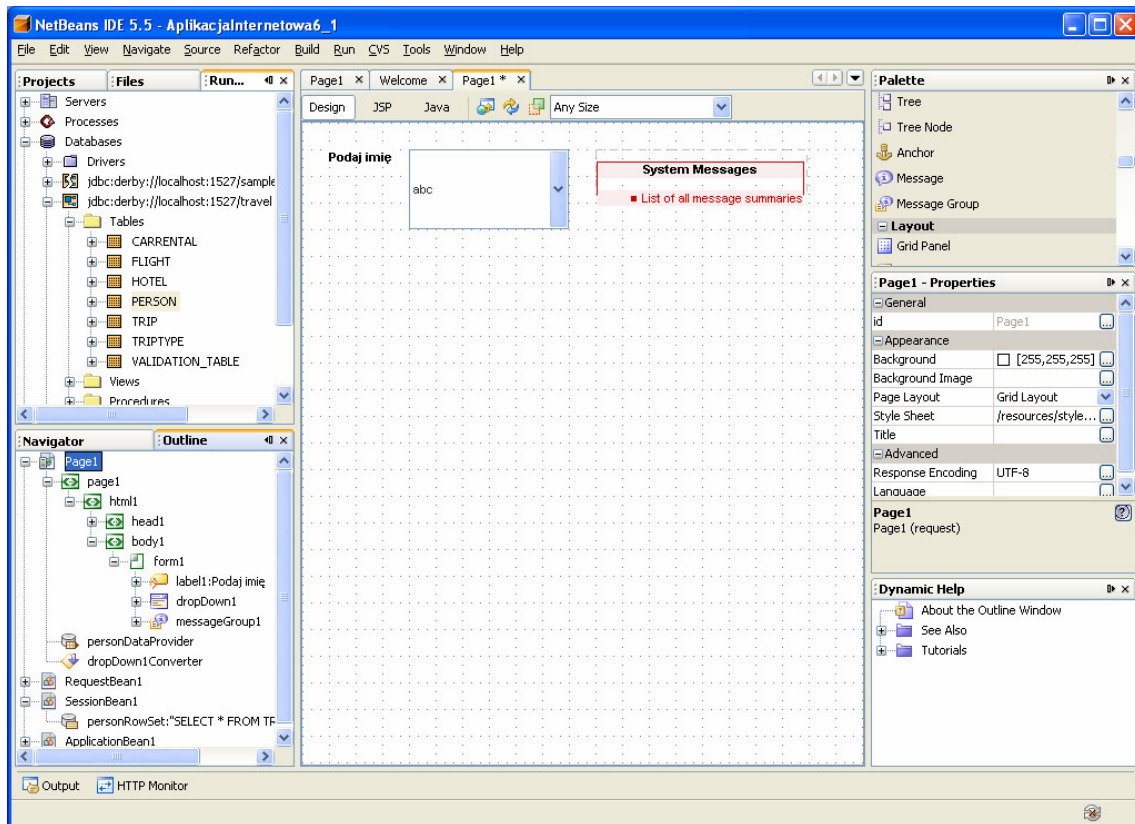
1. Wybierz opcję **File/New Project**. Wybierz kategorię projektu **Web**, a rodzaj projektu **Visual Web Application**
2. Kliknij na **Next**
3. Podaj nazwę projektu (**Project name**), wybierz katalog (**Project Location**)-np.**AplikacjaInternetowa3**.
4. Wybierz serwer aplikacji (**Server**) oraz wersję Javy Enterprise (**Java EE Version**).
5. Okno projektu (**Projects**) zawiera układ plików typu **BluePrints**. Plik **Page1.jsp** jest stroną startową napisaną w języku JSP.
6. Okno **Files** zawiera układ fizyczny plików. Plik **Page1.jsp** oraz **Page1.java** stanowią całość- plik **Page1.java** obsługuje główną stronę jsp.
7. Zaprojektuj stronę **Page1** korzystając z **Palette Basic**
 - 7.1. Przeciągnij komponenty **Label** i **Drop Down List**. W oknach **Properties** obu komponentów nazwy ich są standardowe: **label1** i **dropDown1**. Nazwy te można zmieniać. Na komponencie **Label** napisz np. *Podaj imię*. Naciśnij klawisze **Ctrl+Shift+przeciągnij** utworzoną etykietę **label1** na komponent typu **Drop Down List**. Pole **for** w oknie **Properties** komponentu **label1** zawiera nazwę **dropDown1**.
 - 7.2. Przeciągnij komponent **Message Group**, który będzie wyświetlał komunikaty metod **info(String)**, **error(String)**, **warn(String)**, lub **fatal(String)**. Domyślnie, wyświetlane są komunikaty typu błędy wykonania, błędy walidacji oraz błędy konwersji.



7.3. Połącz komponent `dropDown1` z bazą danych `Travel`. W tym celu należy kliknąć na zakładkę `Runtime`, następnie rozwinąć opcję `Databases` (kliknąć na przycisk `+`) i prawym klawiszem myszy kliknąć na bazę danych `jdbc:derby://localhost:1527/travel[travel on TRAVEL]`. Na wyskakującym menu kliknąć na opcję `Connect`. W kolejnym okienku wpisać hasło `travel` w polu `password`. Po połączeniu z bazą danych uzyskuje się dostęp do tabel (`Tables`), widoków (`Views`) oraz składowanych procedur (`Procedures`) po kliknięciu na „przycisk `+`” bazy danych `Travel`.

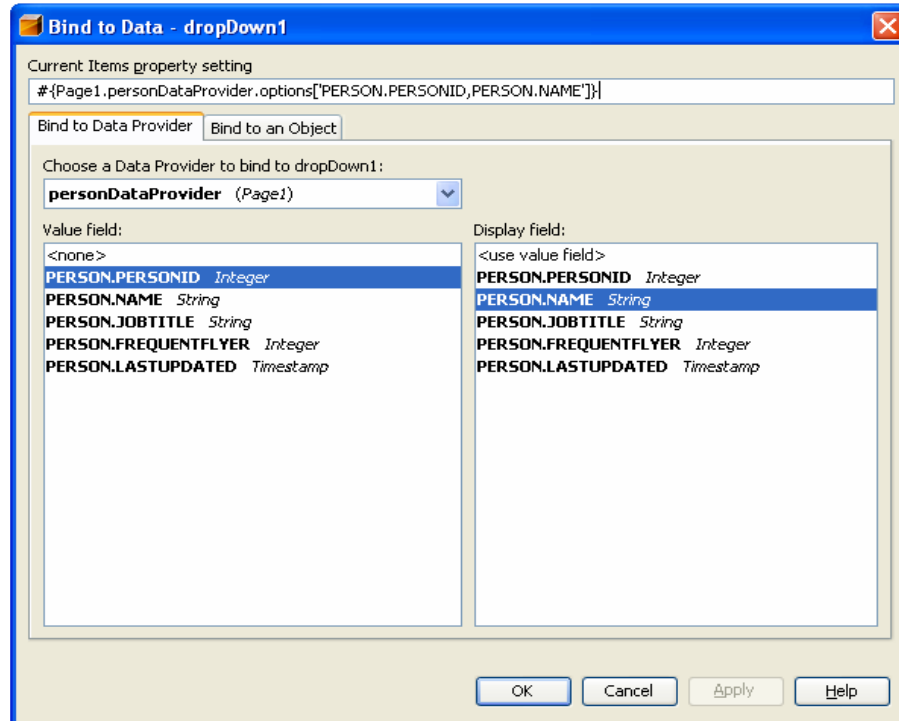


7.4. Przeciągnij tabelę **Person** z zakładki **Runtime** na komponent **dropDown1**. Na komponencie pokazały się elementy listy typu **abc**, co oznacza połączenie z łańcuchami typu **varchar** kolumn wybranej tabeli **Person**.

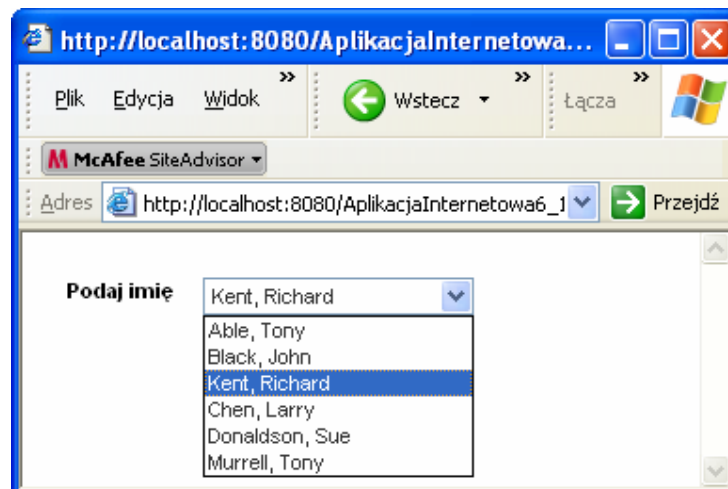


Pojawił się komponent **personDataProvider**, odpowiedzialny za połączenia z warstwa biznesową (obiekty EJB, tablice Arrays) oraz właściwość **personRowSet** w obiekcie **SessionBean1**, odpowiedzialny za połączenie i obsługę zapytań do tabel w bazie danych (wykonanie zapytania oraz zarządzanie jego wynikiem).

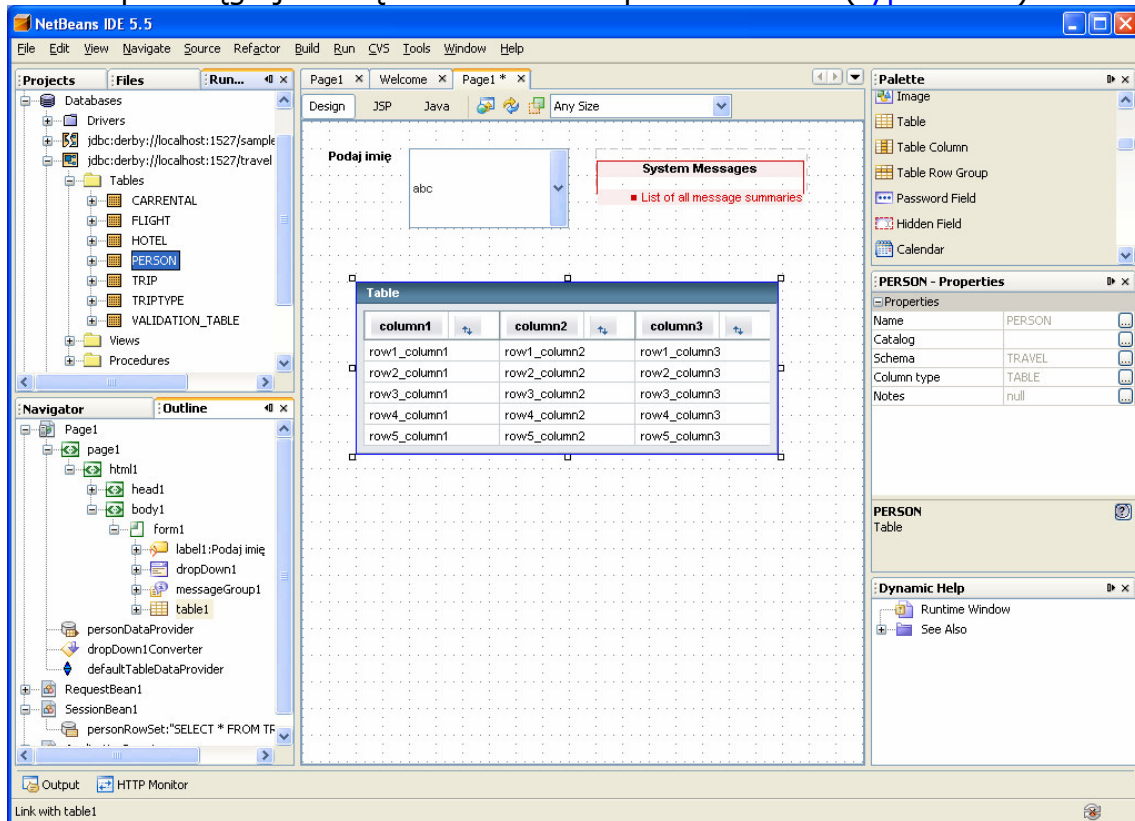
- 7.5. Prawym klawiszem myszy kliknąć na komponencie dropDownList i z wyskakującego menu wybrać **Bind To Data**. W wywołanym formularzu wybrać zakładkę **Value Field** i zaznaczyć **PERSON.PERSONID** jako domyślna wartość w programie oraz w zakładce **Display** wybrać **PERSON.NAME** jako kolumnę do wyświetlania na pozycjach komponentu **dropDown1**.



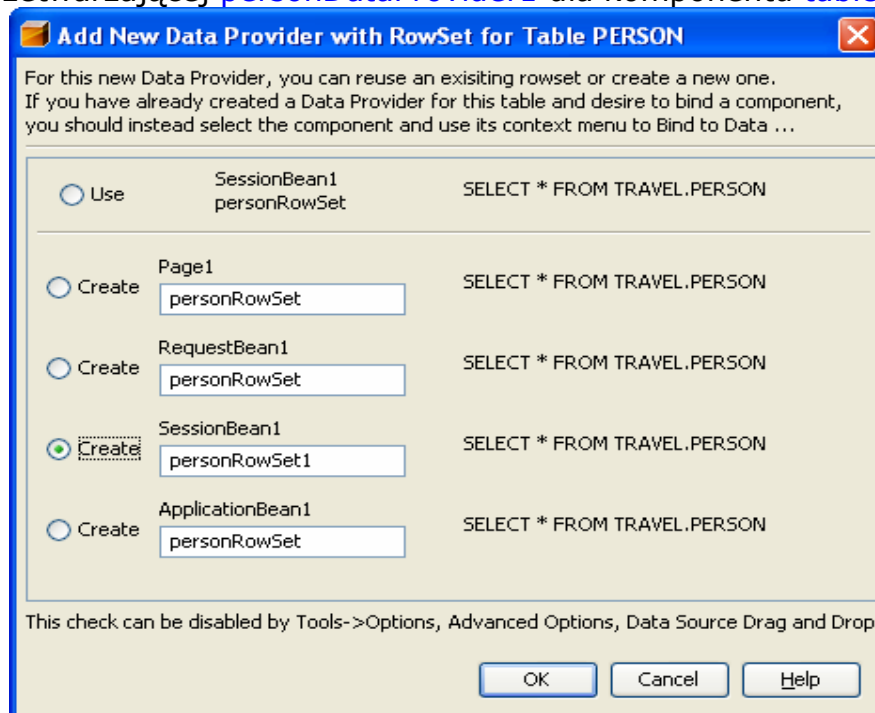
- 7.6. Uruchom aplikację (Kliknij prawym klawiszem myszy w oknie **Project** na nazwę projektu, w ukazanym oknie uruchom kolejno **Build Project**, **Deploy Project**, **Run Project** lub tylko **Run**) i uruchom poszczególne funkcje aplikacji.)



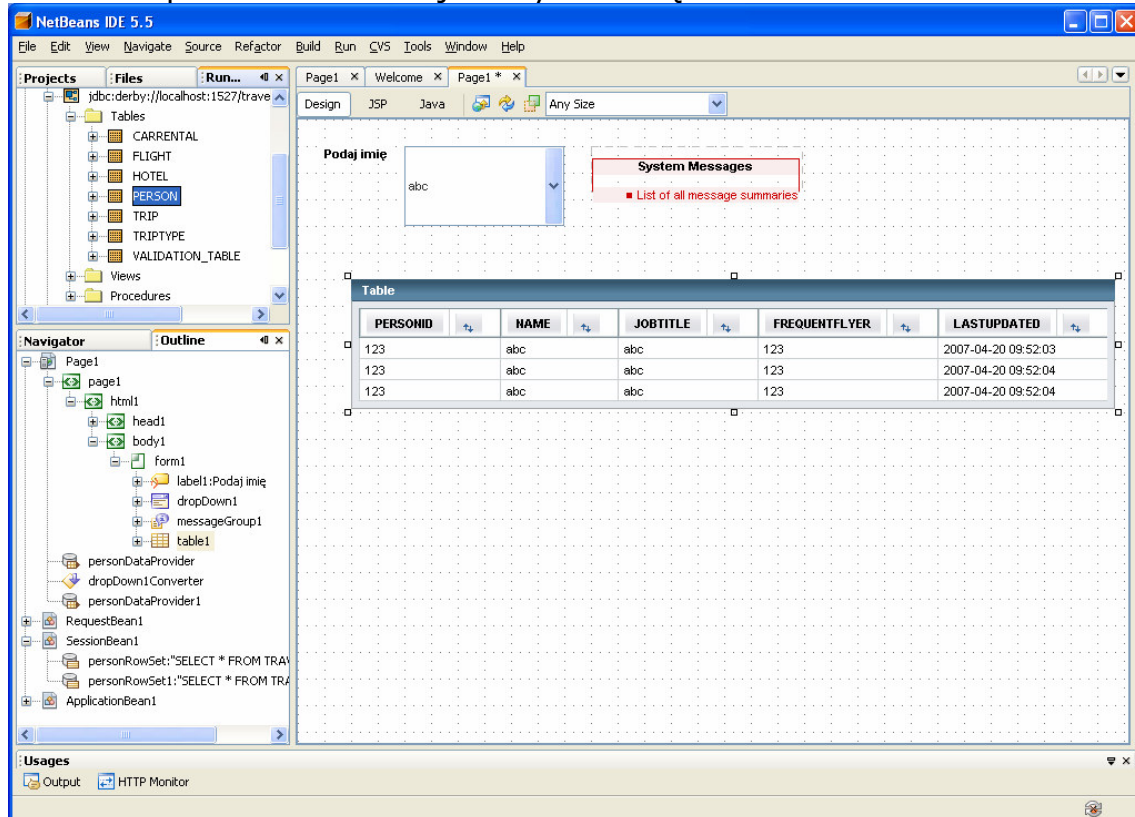
7.7. Przeciągnij komponent **Table** na stronę **Page1** w trybie **Design** i umieść go poniżej komponentu **dropDown1**. W oknie **Runtime** przeciągnij tabelę **Person** na komponent **table1** (typu **Table**).



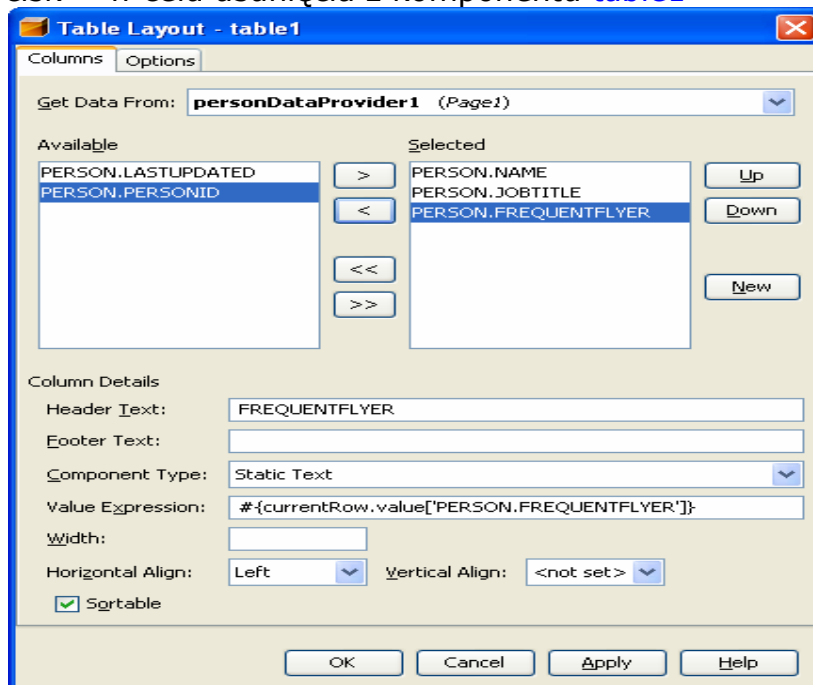
- Wybór nowej warstwy bazodanowej **personRowSet1** oraz warstwy przetwarzającej **personDataProvider1** dla komponentu **table1**

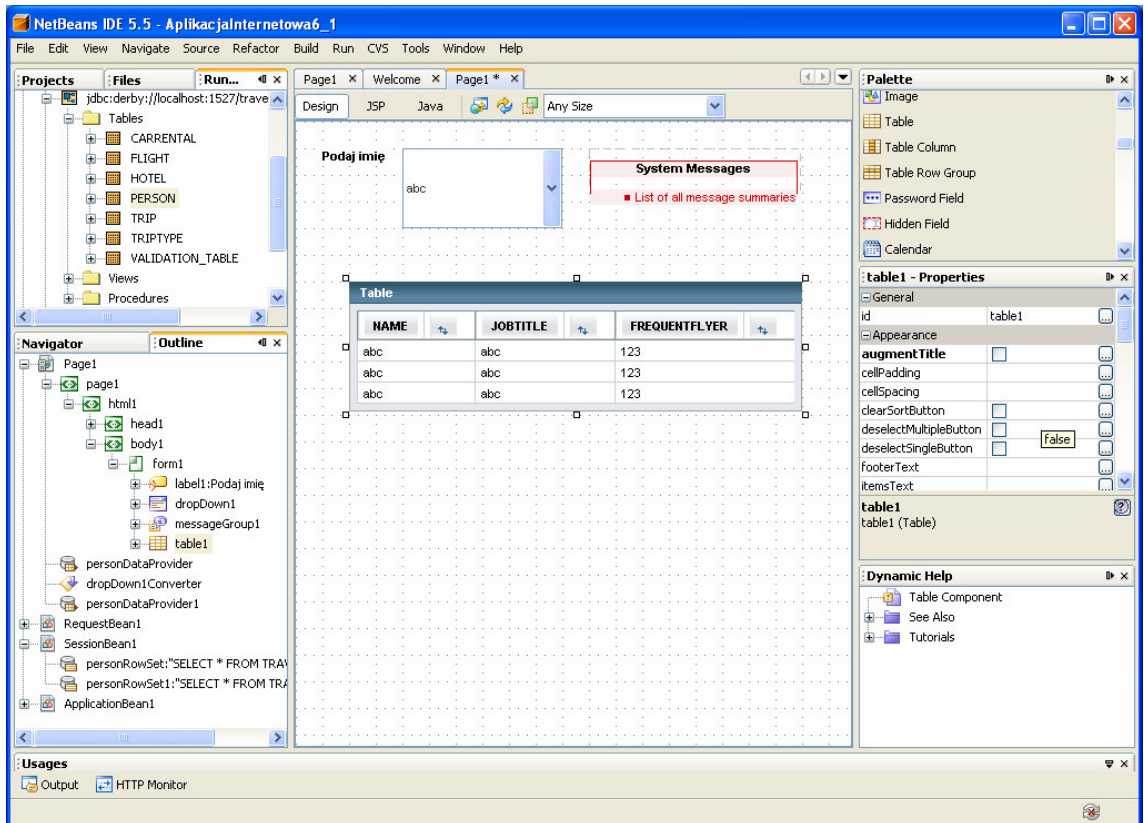


- Komponent `table1` skojarzony z tabelą `PERSON`

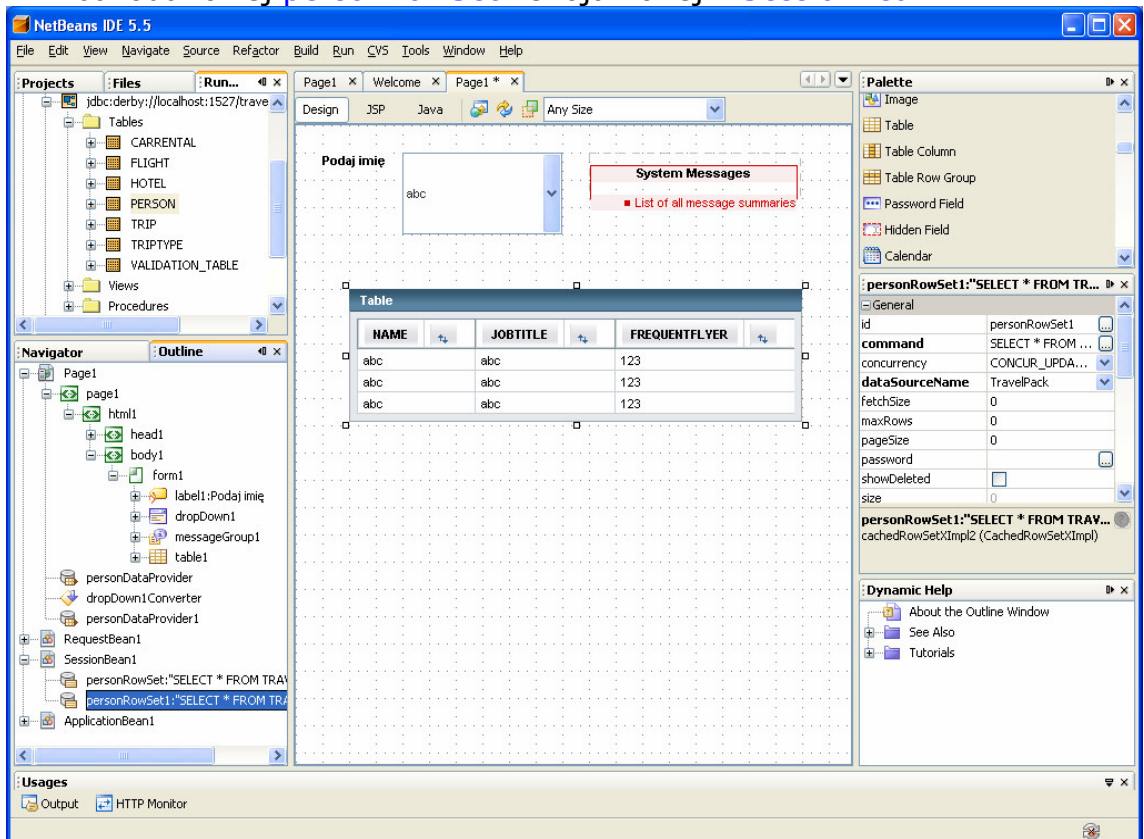


7.8. Kliknij na prawym klawiszem na komponent `table1` i wybierz **Table Layout**. W ukazanym formularzu należy wybrać kolumny tabeli do wyświetlenia. Za pomocą **Ctrl+kliknięcie** wybrać następujące kolumny: `PERSON.LASTUPDATED` i `PERSON.PERSONID` i nacisnąć przycisk **<** w celu usunięcia z komponentu `table1`

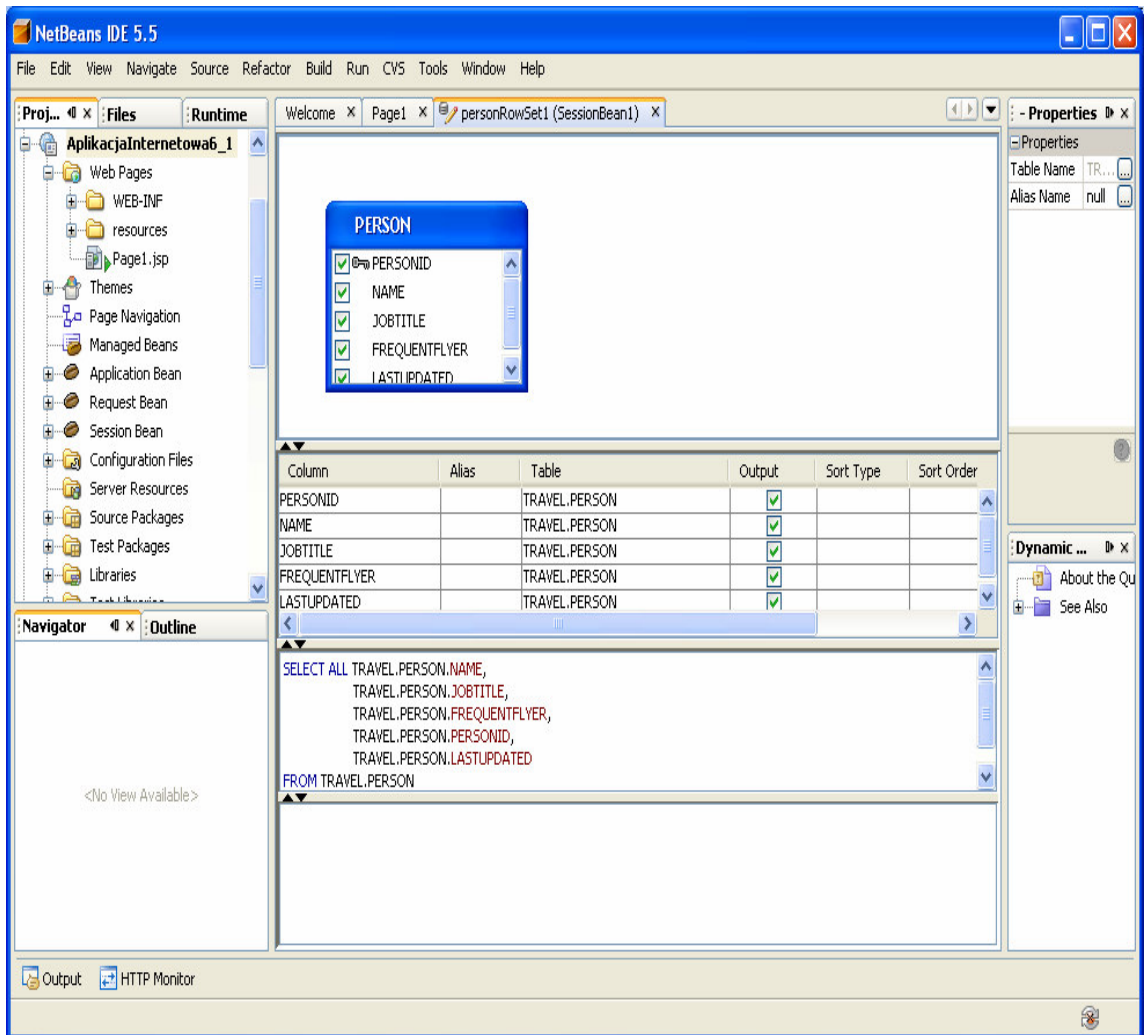




7.9. Podgląd i możliwość edycji zapytania SQL dla warstwy bazodanowej `personRowSet1` skojarzonej z `SessionBean1`



Należy wybrać w okienku OutLine komponent SessionBean1 i wybrać personRowSet1 prawym klawiszem myszy. Z wyskakującego menu wybrać opcję **Edit SQL Statement**.

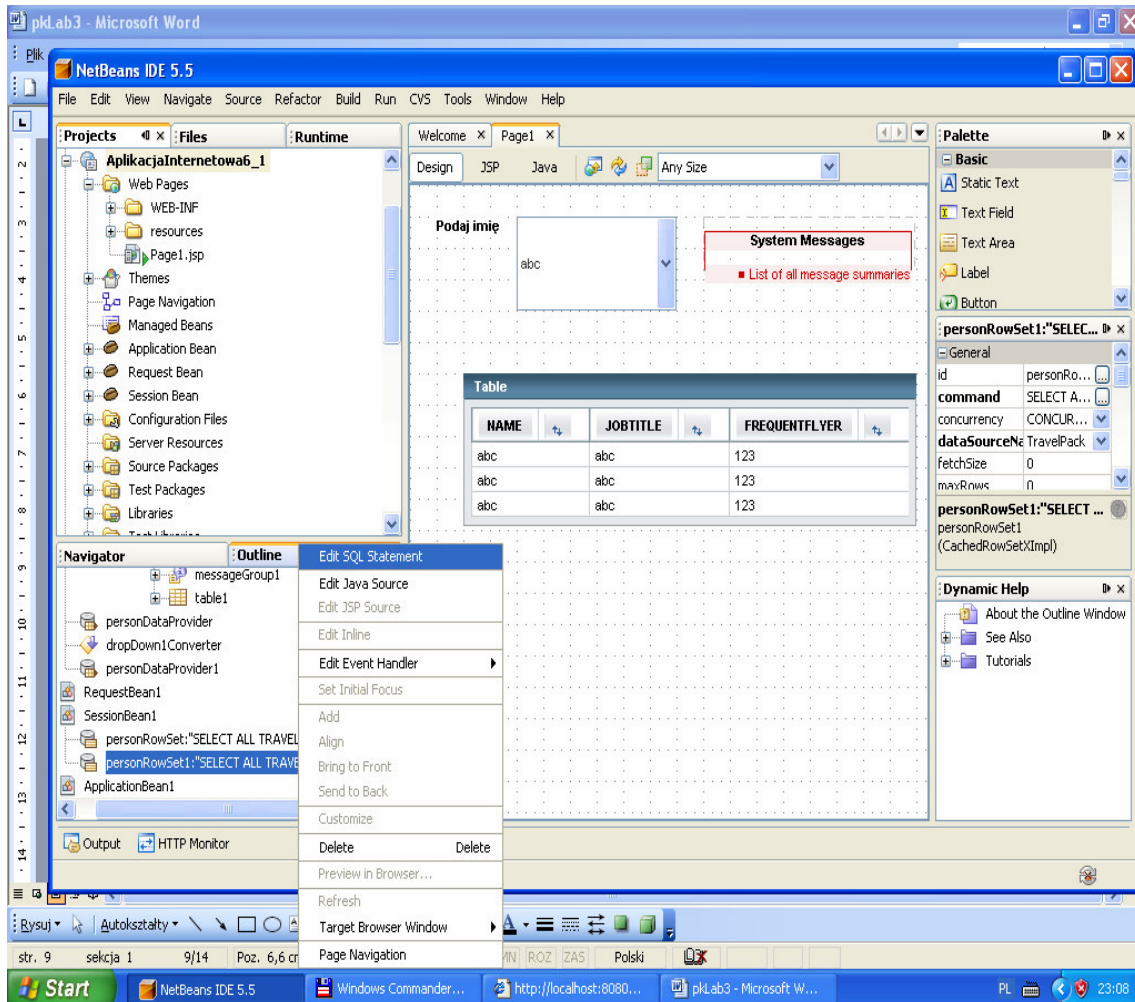


Uwaga1: Mimo, że wyświetlamy jedynie trzy kolumny, nie modyfikujemy zapytania. Można jedynie wyłączyć kolumnę **LASTUPDATED** - można to wykonać w tabeli (część środkowa formularza) ustawiając przycisk typu checkbox na niezaznaczony przy danej kolumnie.

Uwaga: Uruchom aplikację (Kliknij prawym klawiszem myszy w oknie **Project** na nazwę projektu, w ukazanym oknie uruchom kolejno **Build Project**, **Deploy Project**, **Run Project** lub tylko **Run Project**). Po uruchomieniu aplikacji w komponencie table1 są wyświetlane po trzy kolumny (**NAME**, **JOBTITLE**, **FREQUENTFLYER**) wszystkich wierszy tabeli **PERSON**

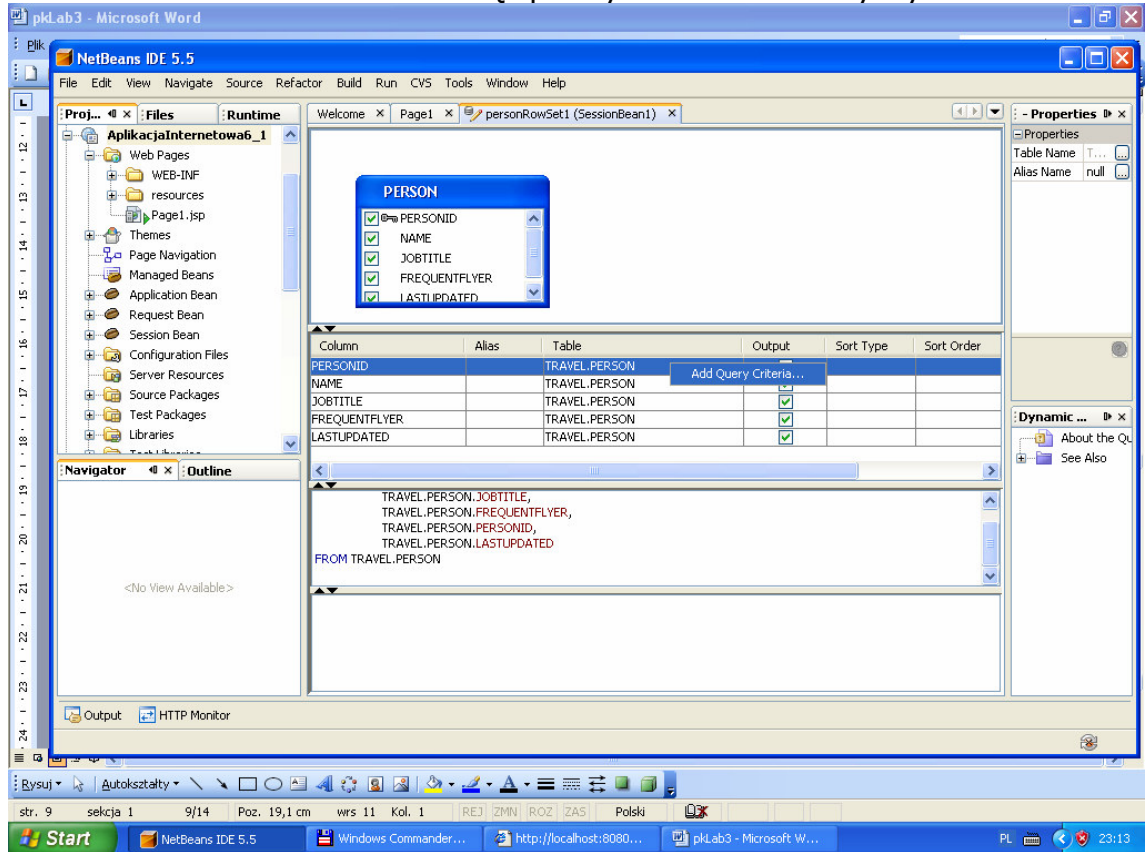
7.10. Ograniczenie wyświetlanych wierszy w tabeli do danych osoby wybranej w komponencie `dropdown1`.

- Należy wybrać `personRowSet1` skojarzony z komponentem `tabel1` prawym klawiszem myszy i wybrać `edytor SQL`

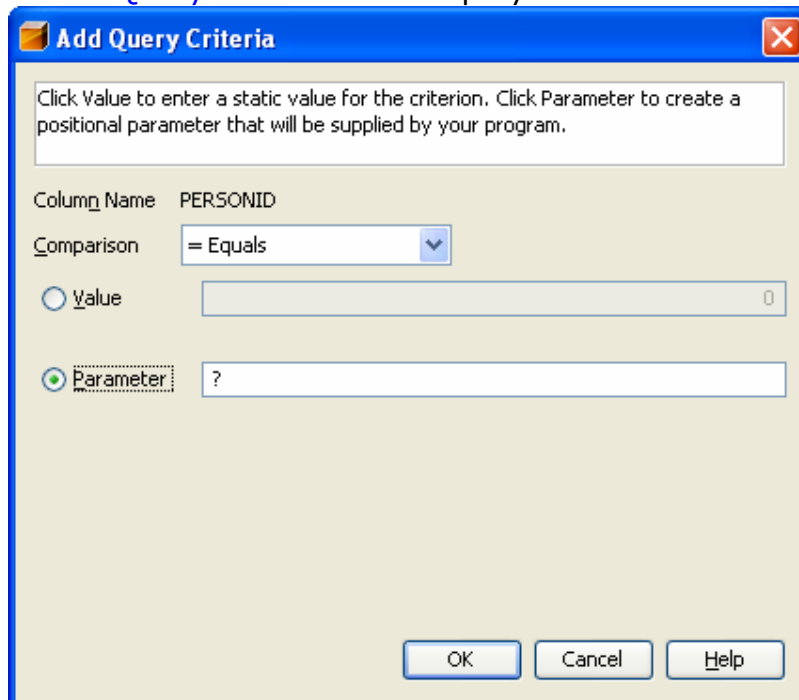


Uwaga2: W tym rozwiązaniu kryterium dla zapytania `Select` jest kolumna `PERSONID`

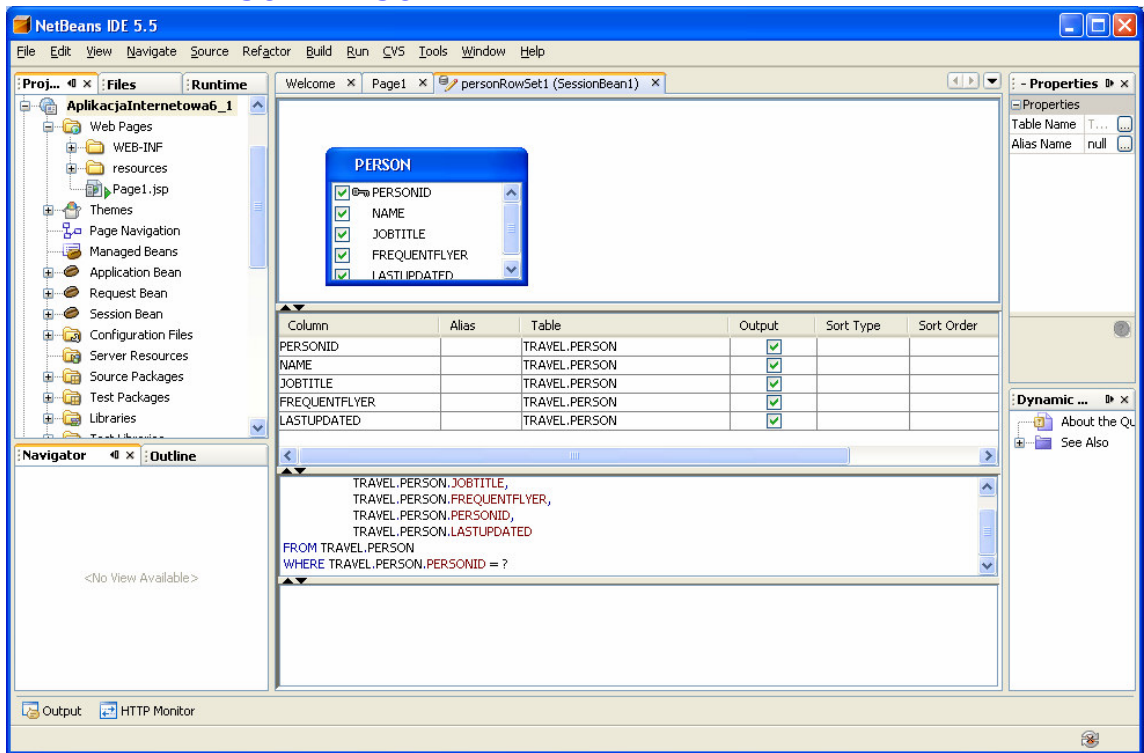
- W kolumnie **PERSONID** kliknąć prawym klawiszem myszy.



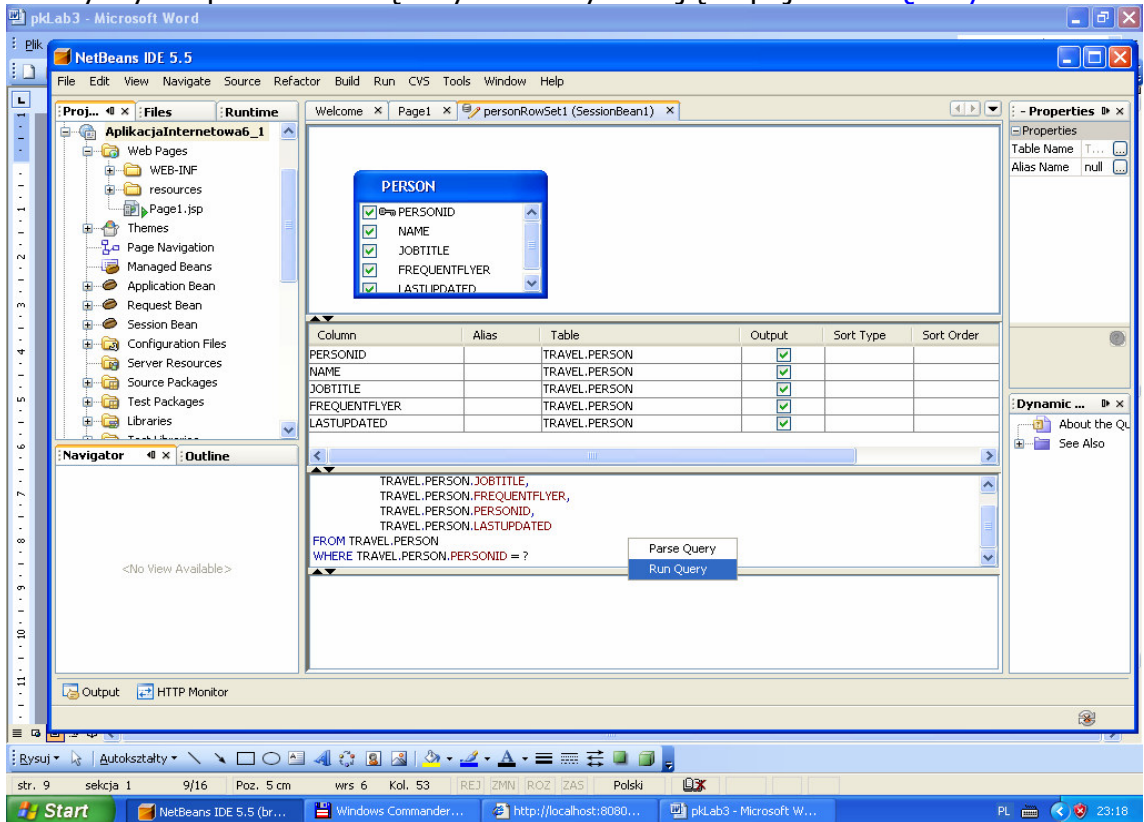
- W menu **Add Query Criteria** ustawić przycisk **Parametr**.



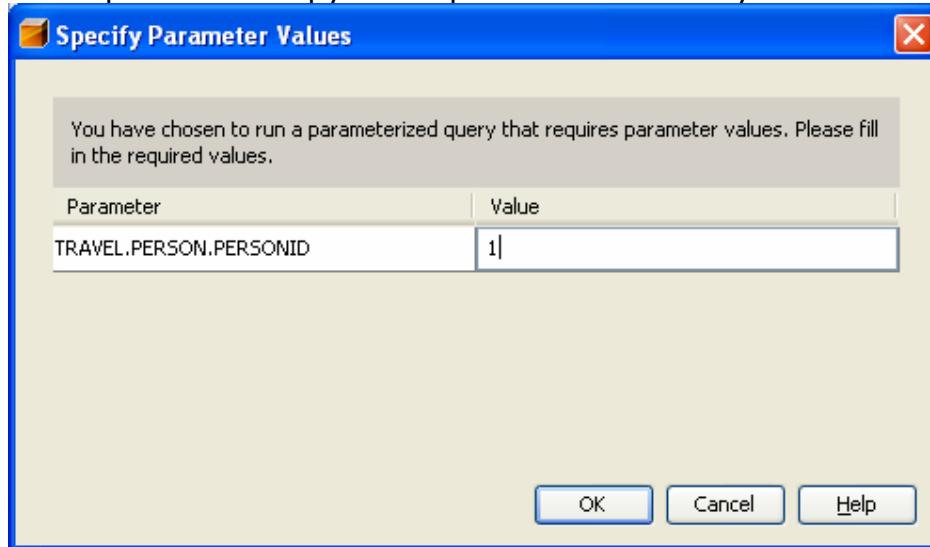
- Zapytanie zostanie zmodyfikowane o warunek **WHERE TRAVEL.PERSON.PERSONID=?**



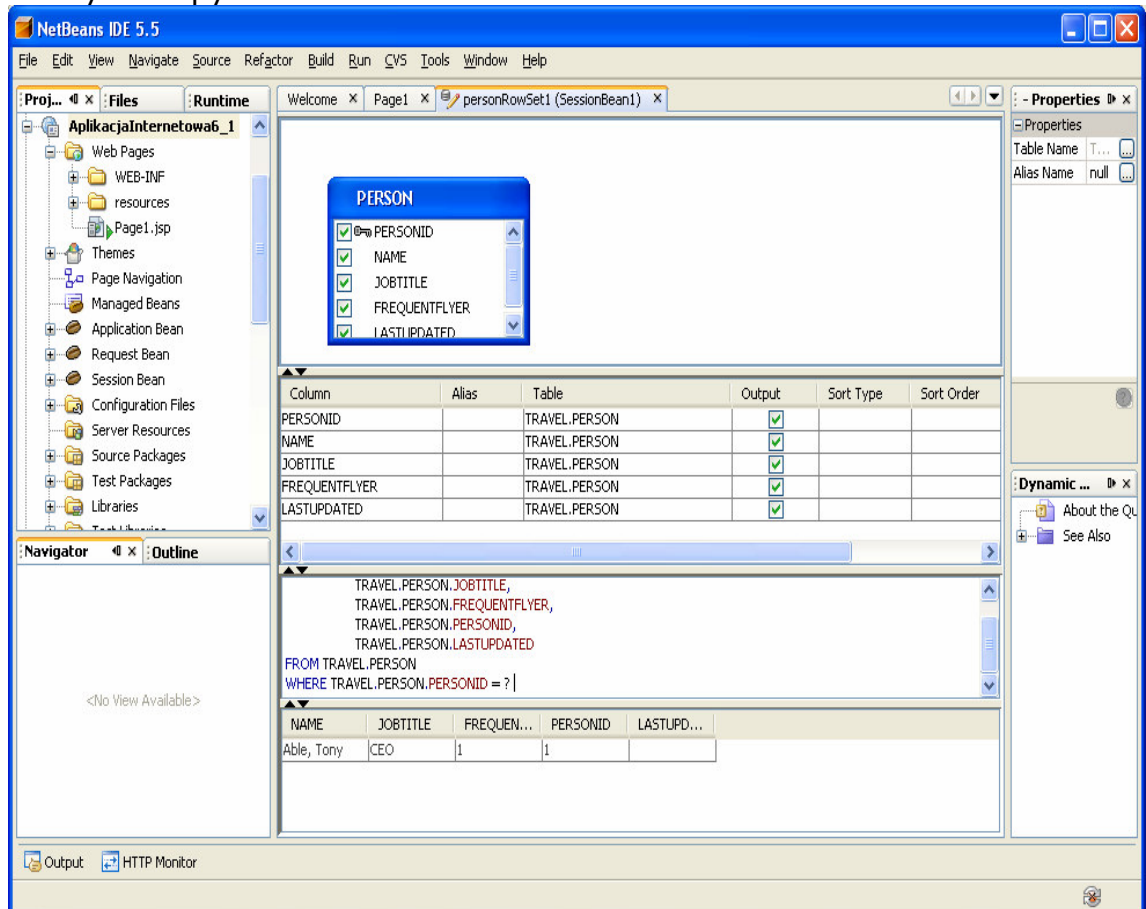
- Można sprawdzić działanie zapytania klikając prawym klawiszem myszy na powierzchnię edytora i wybierając opcje **Run Query**.



- Podanie parametru zapytania np. **PERSONID** równy 1



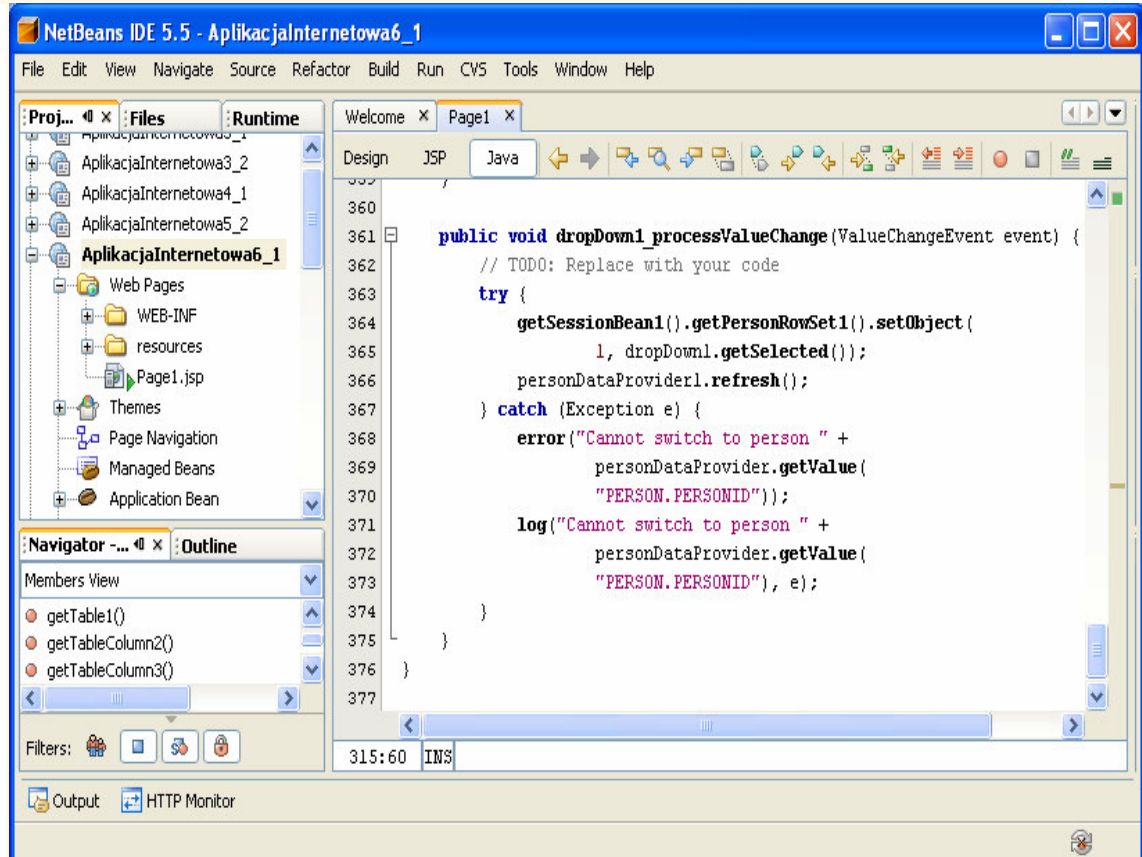
- Wynik zapytania



8. Uzupełnij kod aplikacji w trybie Java strony [Page1](#) (klasa [Page1](#) dziedziczy od klasy [AbstractPageBean](#))

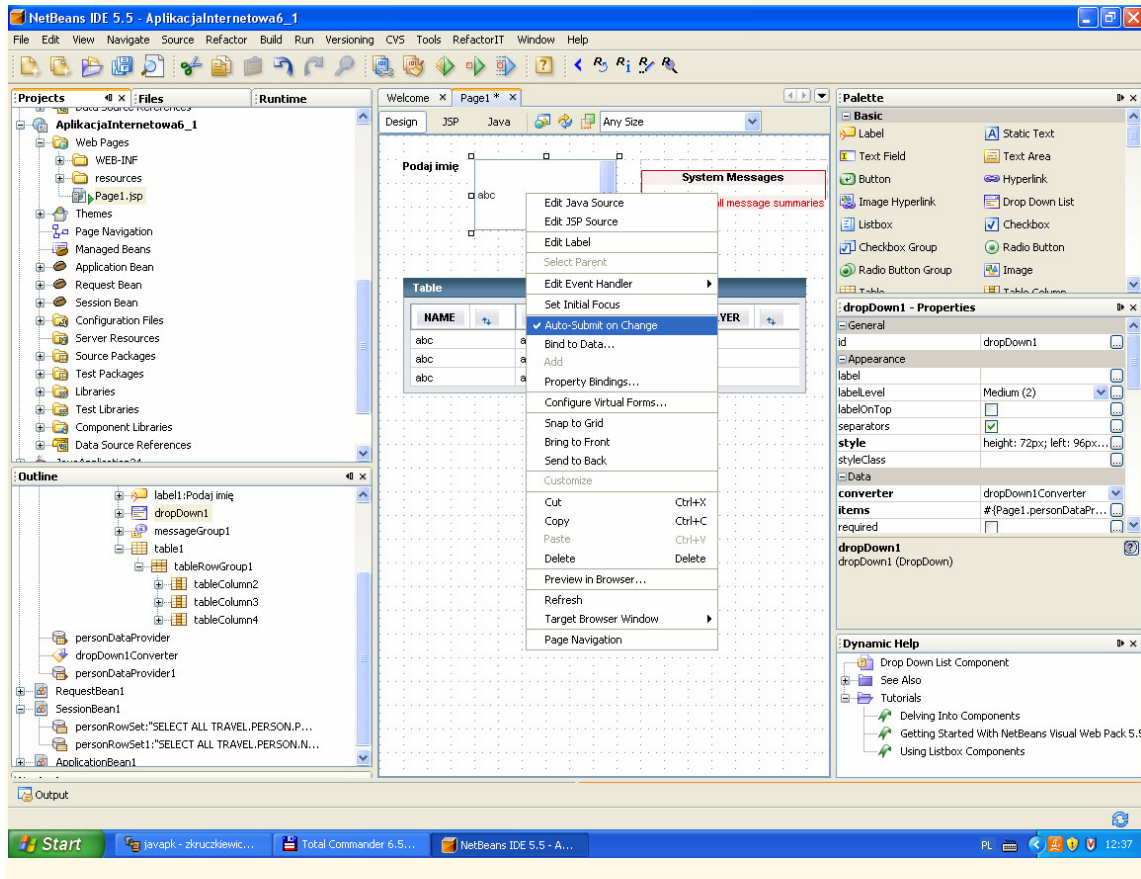
8.1. Należy kliknąć podwójnie na kontrolkę [dropDown1](#) i obsłużyć zdarzenie [ProcessValueChange](#) za pomocą metody [dropDown1_processValueChange](#)

```
public void dropDown1_processValueChange(ValueChangeEvent event)
{
    try { //ustawienie wiersza w komponencie table1 danymi tabeli bazy danych skojarzonej z komponentem dropDown1 i
//wybranymi w tym komponencie
        getSessionBean1().getPersonRowSet1().setObject(
            1, dropDown1.getSelected());
        personDataProvider1.refresh();
    } catch (Exception e) {
        error("Cannot switch to person " +
            personDataProvider1.getValue("PERSON.PERSONID"));
        log("Cannot switch to person " +
            personDataProvider1.getValue("PERSON.PERSONID"), e);
    }
}
```



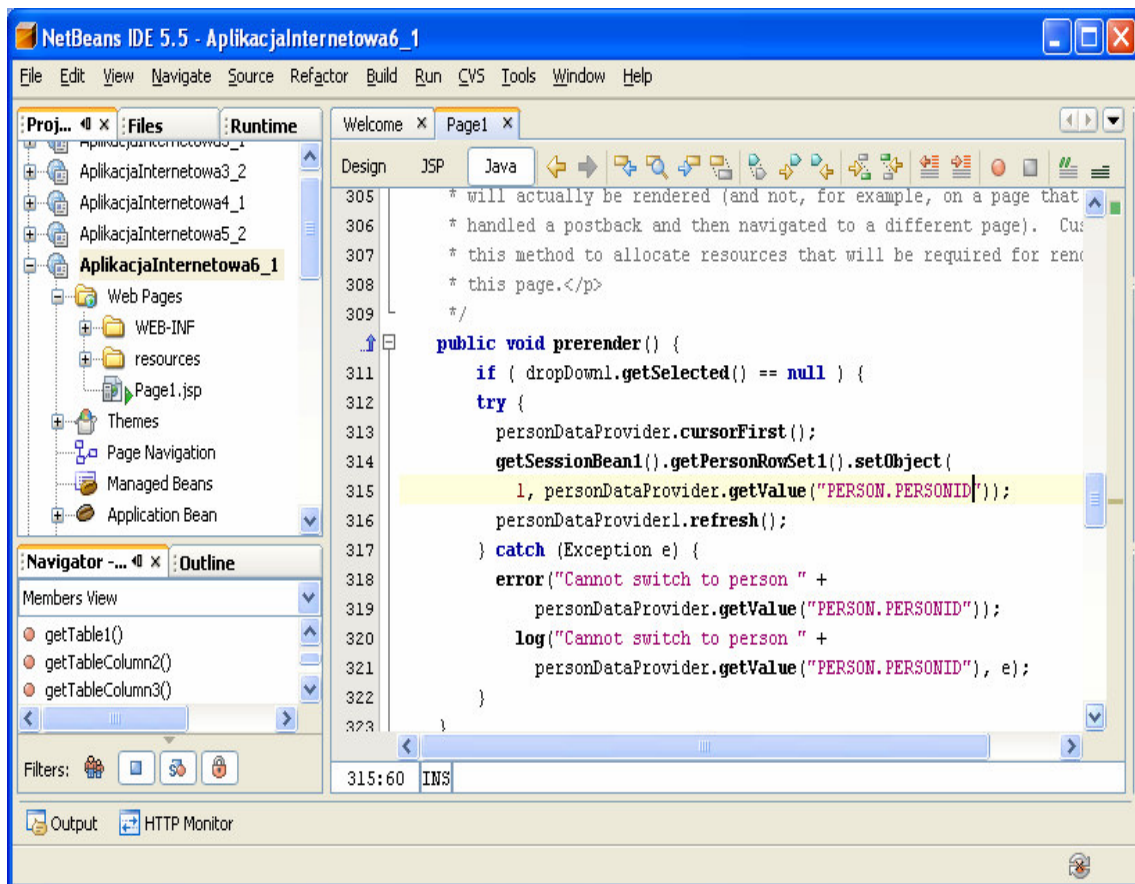
8.2. Przejdź do trybu **Design** strony **Page1**, zaznacz komponent **dropDown1** i prawym klawiszem myszy kliknij nad tym zaznaczonym komponentem. Z wyskakującego menu ustaw **Auto-Submit on Change**.

Pozwoli to obsługiwać zdarzenia wyboru pozycji z listy rozwijanej komponentu **dropDown1**, typu **ProcessValueChange**, za pomocą metody **dropDown1_processValueChange**.



8.3. metoda prerender() //inicjowanie zawartości komponentu `table1`

```
public void prerender()
{
    if ( dropdown1.getSelected() == null ) {
        try { //pobranie pierwszej wiersza z tabeli bazy danych skojarzonej z komponentem dropdown1
            personDataProvider.cursorFirst();
            //ustawienie wiersza komponentu table1 danymi pierwszej krotki tabeli bazy danych skojarzonej z komponentem dropdown1
            getSessionBean1().getPersonRowSet1().setObject
                (1, personDataProvider.getValue("PERSON.PERSONID"));
            personDataProvider1.refresh();
        } catch (Exception e) {
            error("Cannot switch to person " +
                personDataProvider.getValue("PERSON.PERSONID"));
            log("Cannot switch to person " +
                personDataProvider.getValue("PERSON.PERSONID"),e);
        }
    }
}
```



8.3. Uruchom aplikację (Kliknij prawym klawiszem myszy w oknie **Project** na nazwę projektu, w ukazanym oknie uruchom kolejno **Build Project**, **Deploy Project**, **Run Project** lub tylko **Run Project**) i uruchom poszczególne funkcje aplikacji.

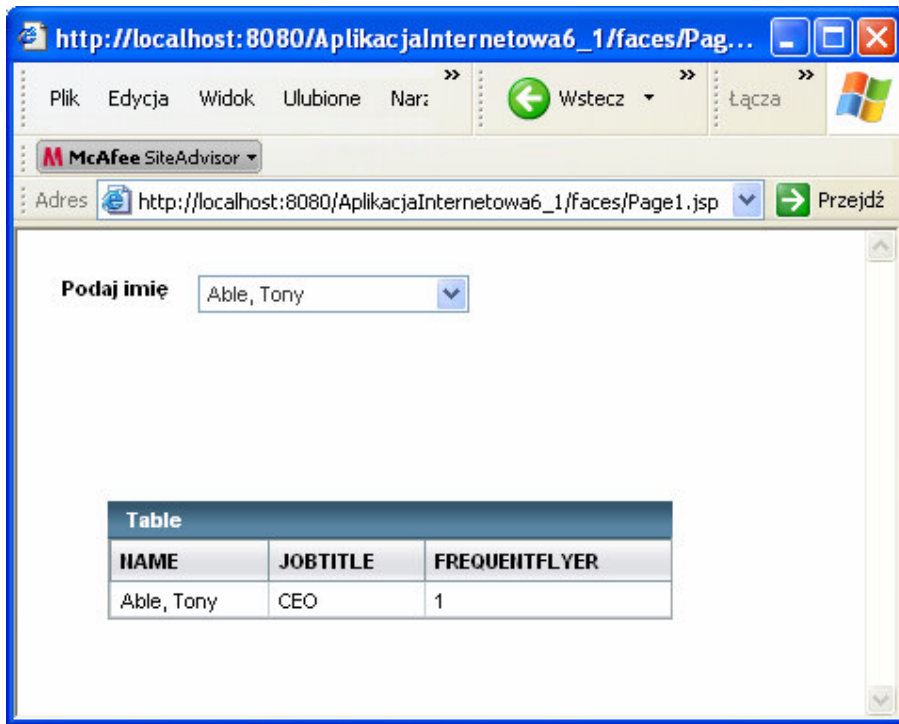


Table 1: Input Components

Component	Description	Palette Section
Text Field	An input field for a single line of text.	Basic
Text Area	An input field for multiple lines of text.	Basic
Password Field	An input field that echoes the input characters with a replacement character to mask the input.	Basic
Calendar	An input field with a pop-up calendar that allows the user to select a date.	Basic
Drop Down List	A drop-down menu, also referred to as a combo box.	Basic
Listbox	A list from which the user can select either one item or multiple items, depending on how the component is configured.	Basic
Checkbox	A single-character box that the user can either select (check) or clear.	Basic
Radio Button	A single radio button that the user can select (check).	Basic
Add Remove List	Two lists (one for available options, one for selected options) with buttons to move the options between the lists, and to order the selected options.	Composite
File Upload	A component with a text input field and a Browse button that displays a file chooser for specifying a file to upload. The application uploads the specified file when the user submits the page.	Basic

Table 2: Display Components

Component	Description	Palette Section
Static Text	Field for displaying text.	Basic
Label	Text field that can be associated with an input field and for which you can specify a weak, medium, or strong font style.	Basic
Image	Inline image.	Basic
Message	Text field that is linked to a specific component for displaying validation errors and other messages about that component.	Basic
Message Group	Text field for displaying runtime error messages, program generated error messages, and, optionally, validation errors and other messages about components that are on the page.	Basic
Alert	Displays an icon and informational text such as a warning, an error, or the successful completion of some event.	Composite
Page Alert	Similar to an Alert component, but is intended for displaying the icon and information on a separate page.	Layout

Table 3: Grouping Components

Component	Description	Palette Section
-----------	-------------	-----------------

Checkbox Group	Displays two or more check boxes in a grid layout.	Basic
Radio Button Group	Displays two or more radio buttons in a grid layout and ensures that only one button can be selected at a time.	Basic
Table, Table Row Group, and Table Column	Displays data from a composite data type such as a database table or an array.	Basic
Tree and Tree Node	Renders an expandable list in a tree structure.	Basic
Tab Set and Tab	Displays different layouts on the same page. Also can be used as a navigational tool.	Layout
Grid Panel	Organizes the components within a layout of rows and columns.	Layout
Group Panel	Groups a set of components in flow layout mode.	Layout
Layout Panel	Use to group a set of components in flow layout mode or grid layout mode.	Layout
Property Sheet, Property Sheet Section, and Property	Lays out a single column of labeled components quickly, and divides the components into sections.	Layout
Breadcrumbs	Lays out a series of link components separated by right angle brackets (>).	Composite
Page Fragment Box	Groups components that you want to consistently display on two or more pages.	Layout

Table 4: Action Components

Component	Description	Palette Section
Button	Button that submits the associated form.	Basic
Hyperlink	Text field that submits a URL or submits a form.	Basic
Image Hyperlink	Image that submits a URL or submits a form.	Basic
Tree Node	Subcomponent of a Tree or Tree Node. A leaf tree node can optionally submit a URL or submit a form.	Basic
Tab	Subcomponent of a Tab Set or a Tab. A tab can optionally submit a URL or submit a form.	Composite