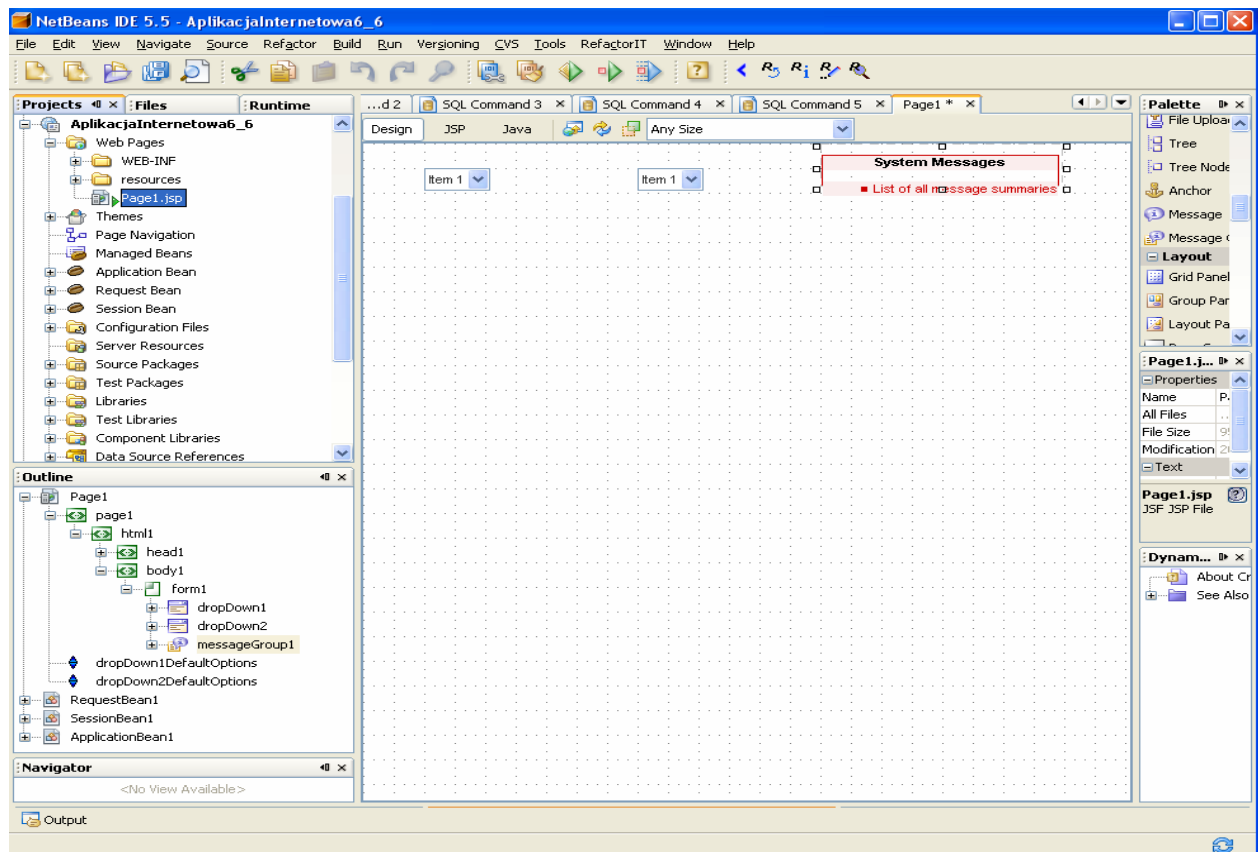


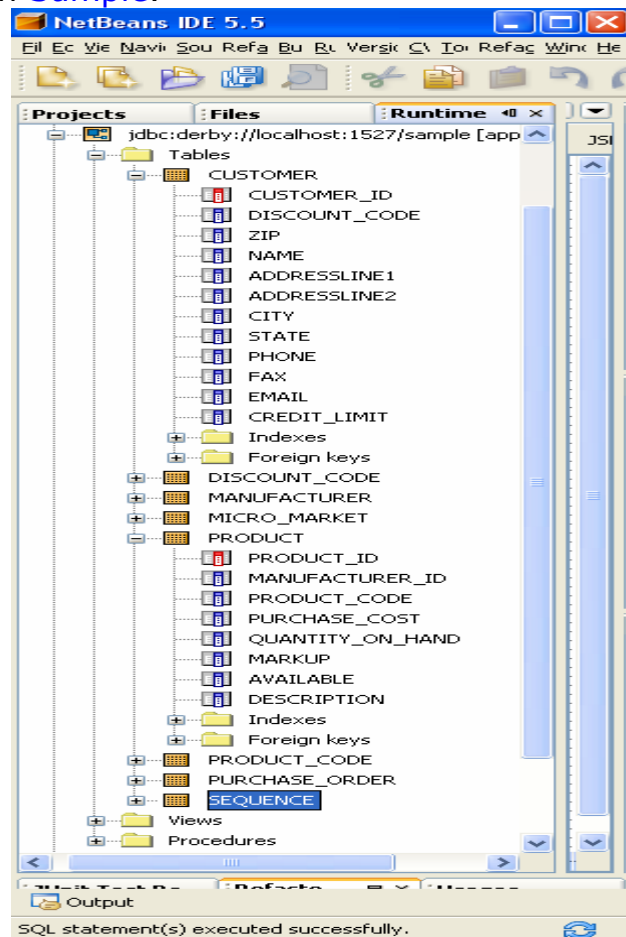
Lab6.

Dwie tabele powiązane relacją jeden do wiele do jednej tabeli (porównaj laboratorium 3, laboratorium 5) – **Baza danych Sample: tabela Customer w relacji jeden do wiele do tabeli PURCHASE_ORDER oraz tabela PRODUCT w relacji jeden do wiele do tabeli PURCHASE_ORDER**

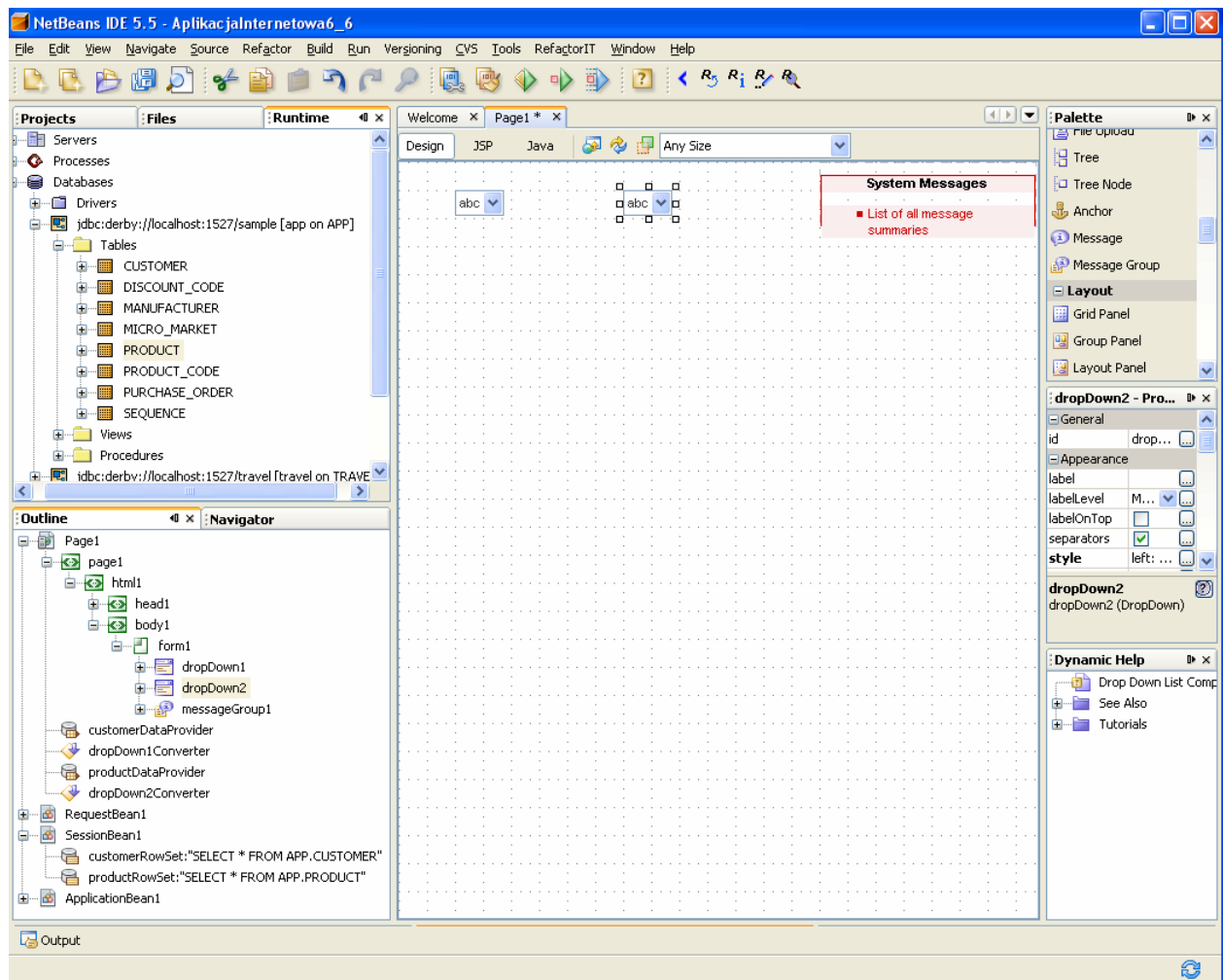
1. Wybierz opcję **File/New Project**. Wybierz kategorię projektu **Web**, a rodzaj projektu **Visual Web Application**
2. Kliknij na **Next**
3. Podaj nazwę projektu (**Project name**), wybierz katalog (**Project Location**)- np. **AplikacjaInternetowa6_6**
4. Wybierz serwer aplikacji (**Server**) oraz 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 dwa komponenty **Drop Down List**. W oknach **Properties** obu komponentów nazwy ich są standardowe: **dropDown1** i **dropDown2**. Nazwy te można zmieniać.
 - 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 `Sample`. 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/sample [app on App]`. Na wyskakującym menu kliknąć na opcję `Connect`. W kolejnym okienku wpisać hasło `app` 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 `Sample`.

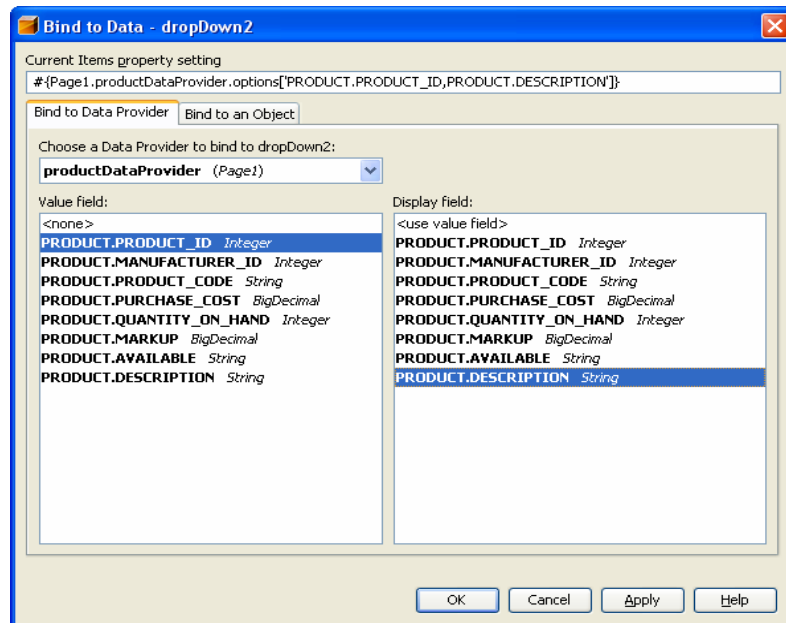
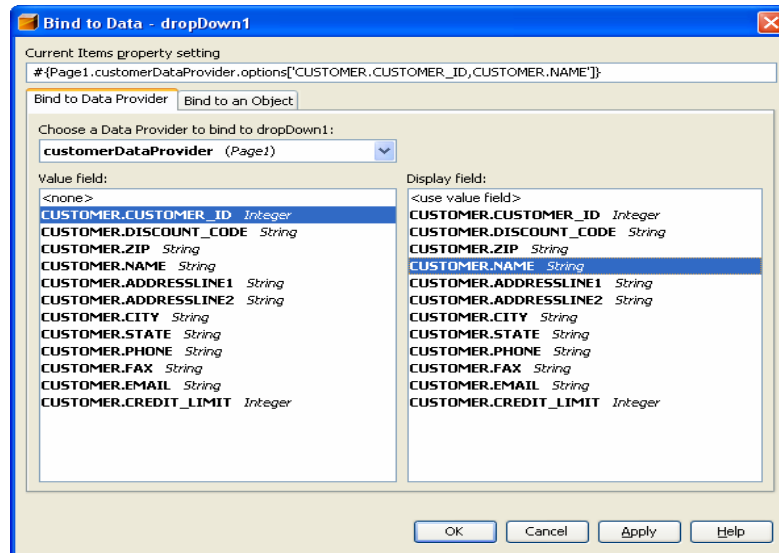


- 7.4. Przeciągnij tabelę `Customer` z zakładki `Runtime` na komponent `dropDown1`. Na komponencie pokazały się elementy listy typu `abc`, co oznacza połączenie z łańcuchami typu `varchar` kolumn wybranej tabeli `Customer`. Przeciągnij tabelę `PRODUCT` z zakładki `Runtime` na komponent `dropDown2`. Na komponencie pokazały się elementy listy typu `abc`, co oznacza połączenie z łańcuchami typu `varchar` kolumn wybranej tabeli `PRODUCT`.

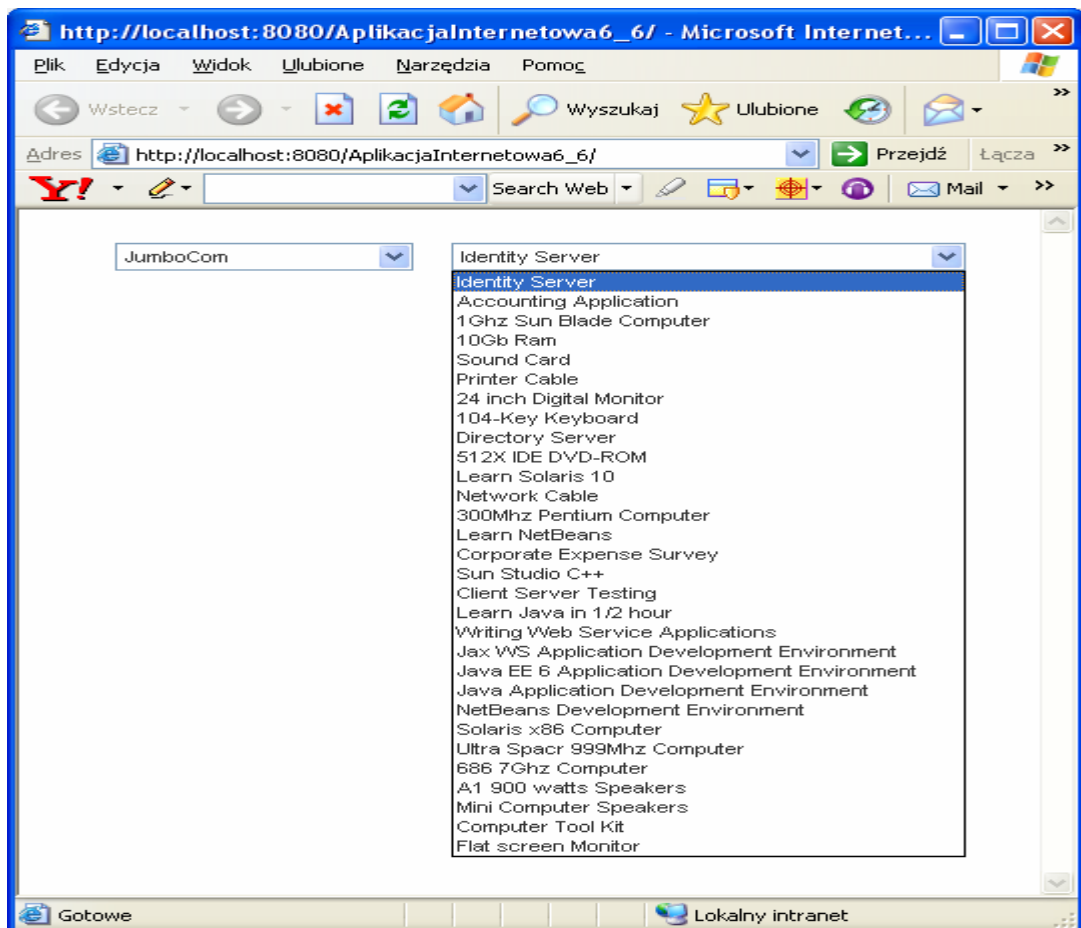
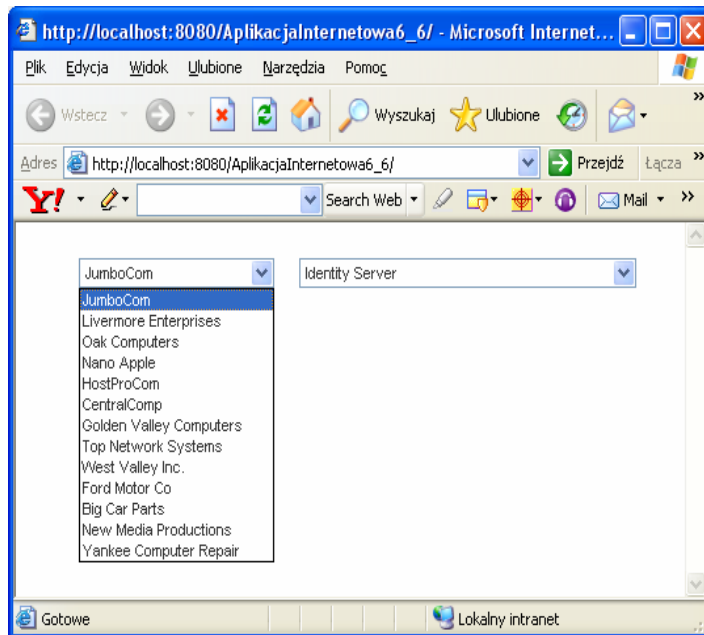


Pojawiły się komponenty [customerDataProvider](#) i [productDataProvider](#), odpowiedzialne za połączenia z warstwą biznesową (obiekty EJB, tablice Arrays) oraz właściwości [customerRowSet](#) i [productRowSet](#) w obiekcie [SessionBean1](#), odpowiedzialne za połączenie i obsługę zapytań do tabel w bazie danych (wykonanie zapytania oraz zarządzanie jego wynikiem).

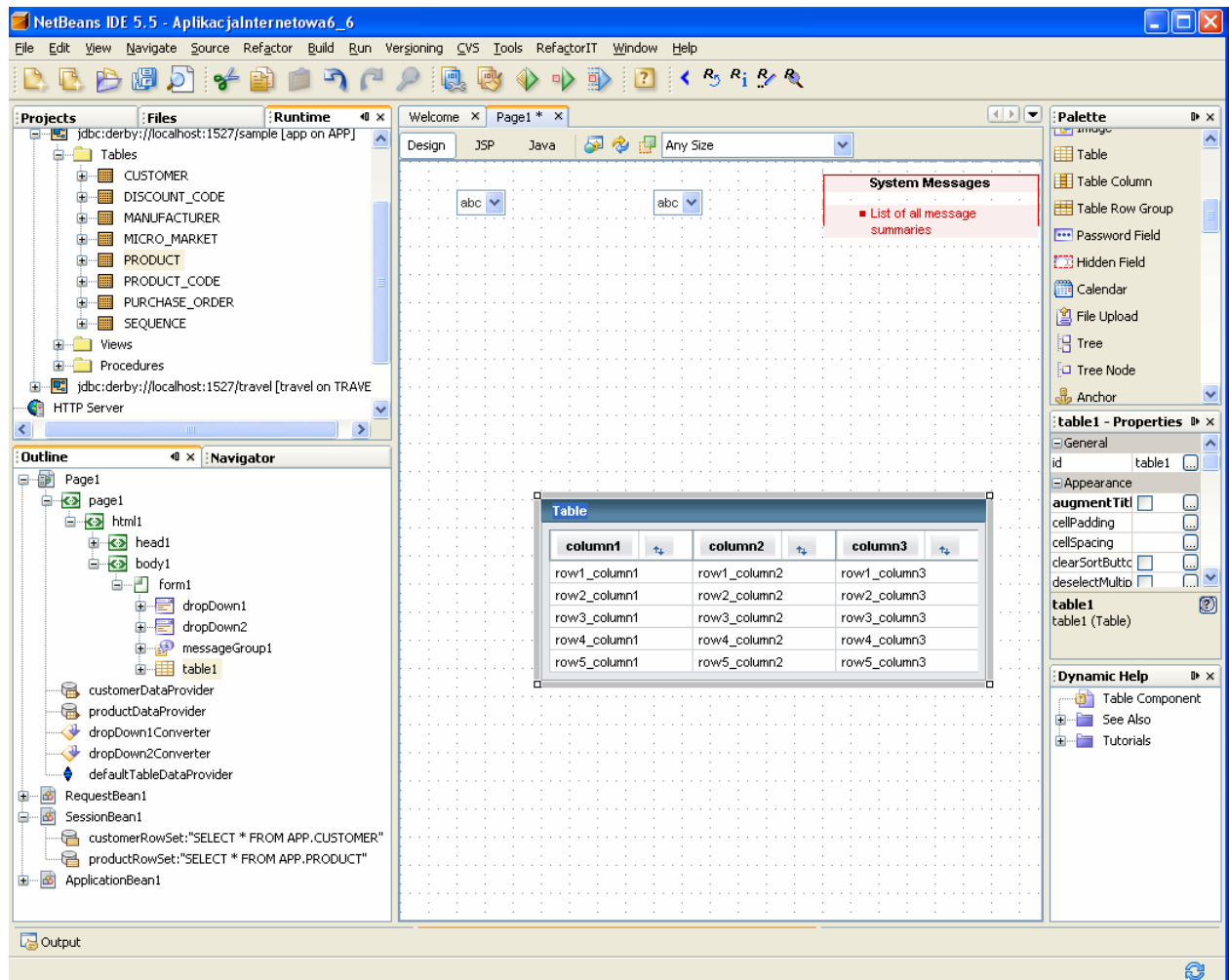
7.5. Prawym klawiszem myszy kliknąć na komponentcie [dropDown1](#) i z wyskakującego menu wybrać [Bind To Data](#). W wywołanym formularzu wybrać zakładki [Value](#) oraz [Field](#) i zaznaczyć [CUSTOMER.CUSTOMER_ID](#) jako domyślna wartość w programie oraz w zakładce [Display](#) wybrać [CUSTOMER.NAME](#) jako kolumnę do wyświetlania na pozycjach komponentu [dropDown1](#). Następnie prawym klawiszem myszy kliknąć na komponentcie [dropDown2](#) i z wyskakującego menu wybrać [Bind To Data](#). W wywołanym formularzu wybrać zakładki [Value](#) oraz [Field](#) i zaznaczyć [PRODUCT.PRODUCT_ID](#) jako domyślna wartość w programie oraz w zakładce [Display](#) wybrać [PRODUCT.DESCRPTION](#) jako kolumnę do wyświetlania na pozycjach komponentu [dropDown2](#).



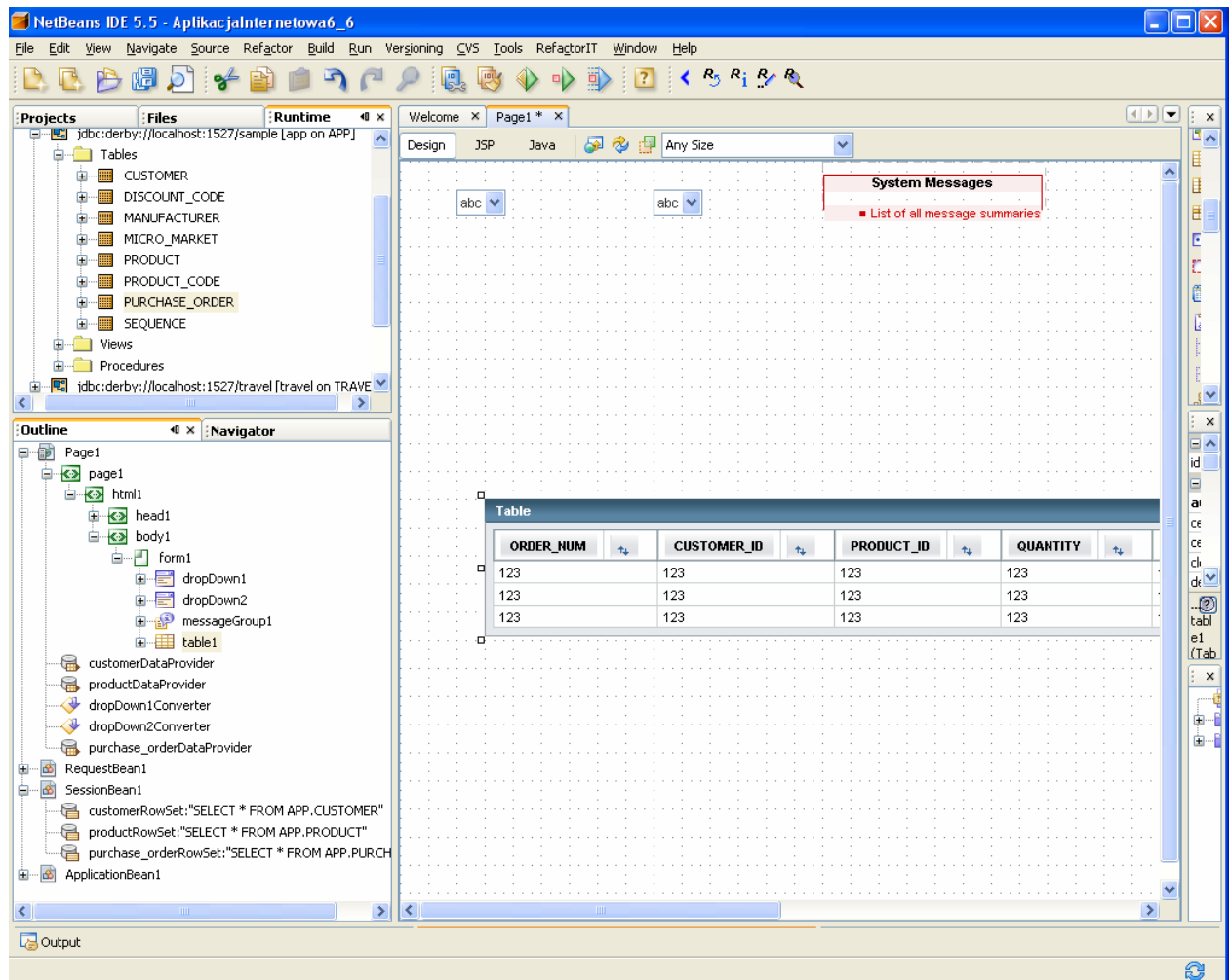
7.6. Uruchom aplikację (Kliknij prawym klawiszem myszy w oknie [Project](#) na nazwę projektu, w ukazanym oknie uruchom kolejno [Build Project](#), [Deploy Project](#), [Run Project](#) lub tylko [Run](#)) i uruchom poszczególne funkcje aplikacji.)



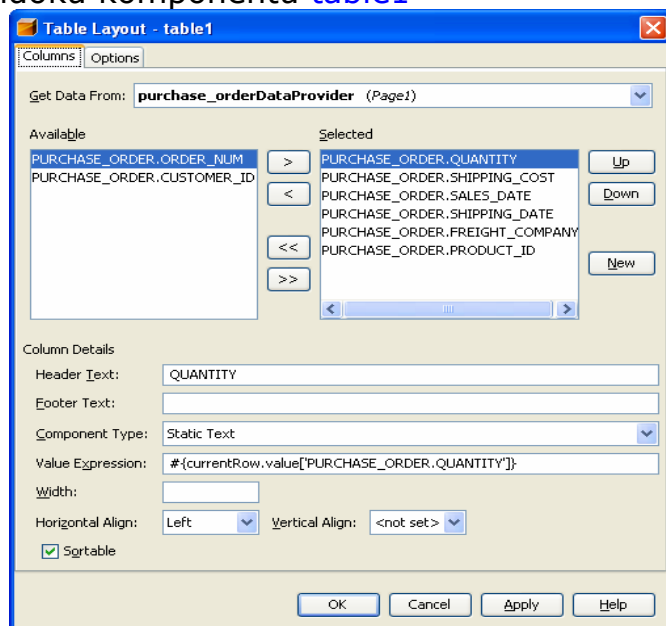
7.7. Przeciągnij komponent [Table](#) na stronę [Page1](#) w trybie [Design](#) i umieść go poniżej komponentów [dropDown1](#) i [dropDown2](#).

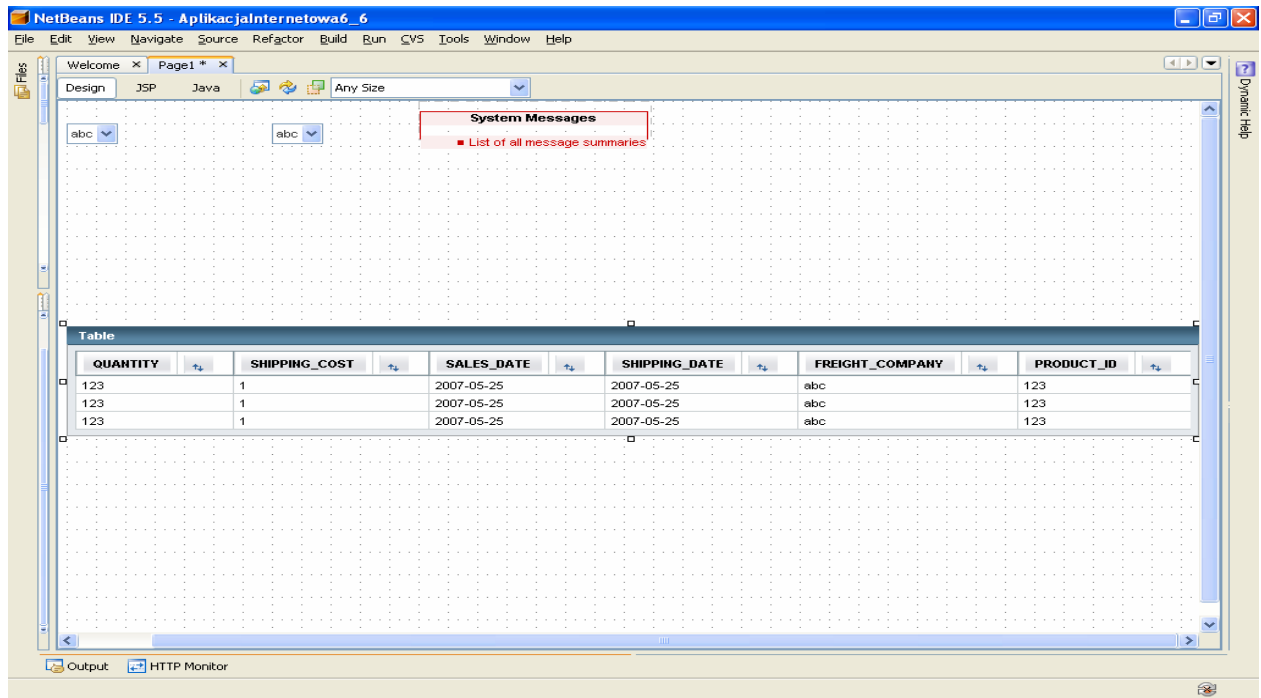


W oknie Runtime przeciągnij tabelę PURCHASE_ORDER na komponent table1 (typu Table).

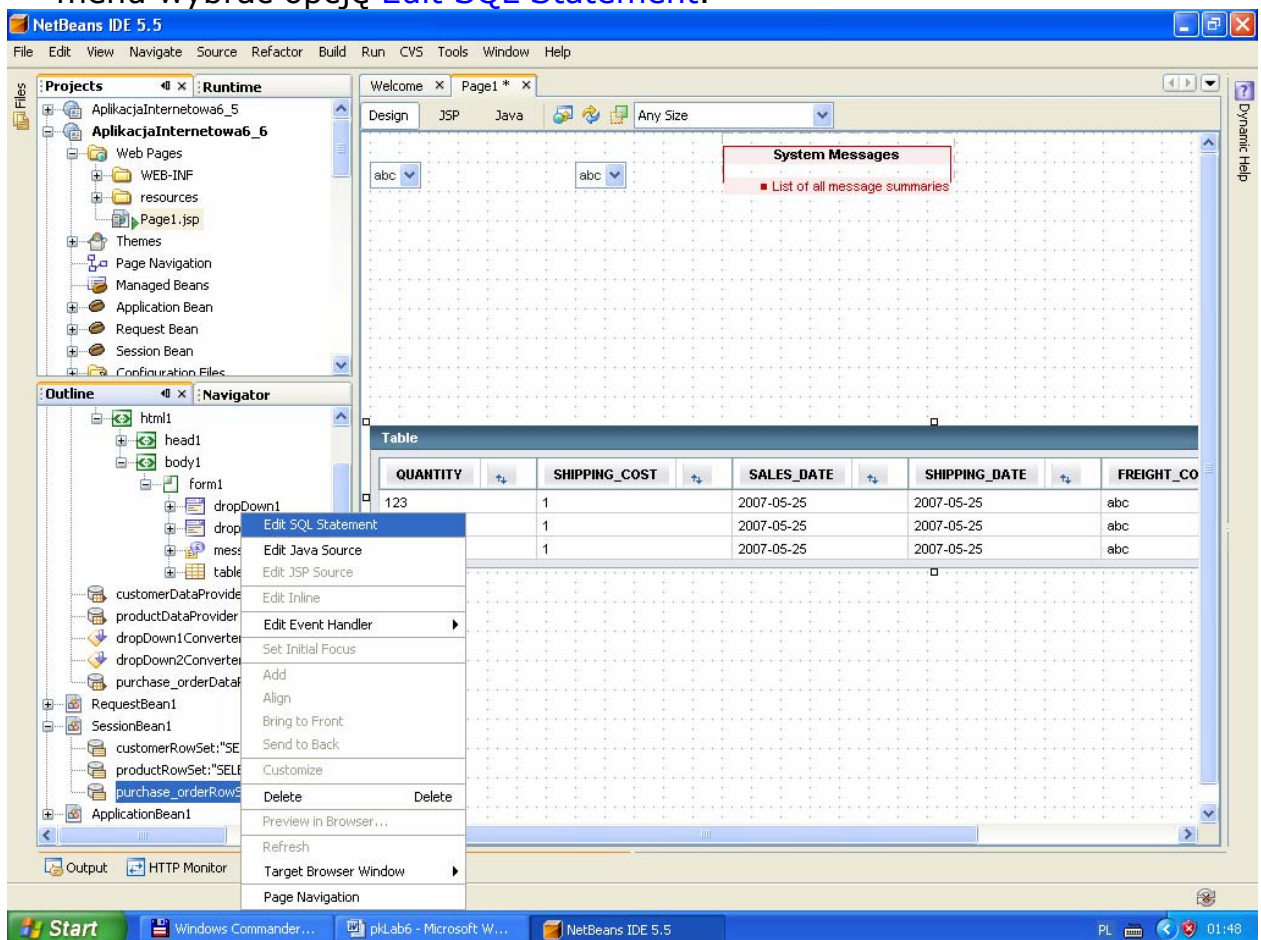


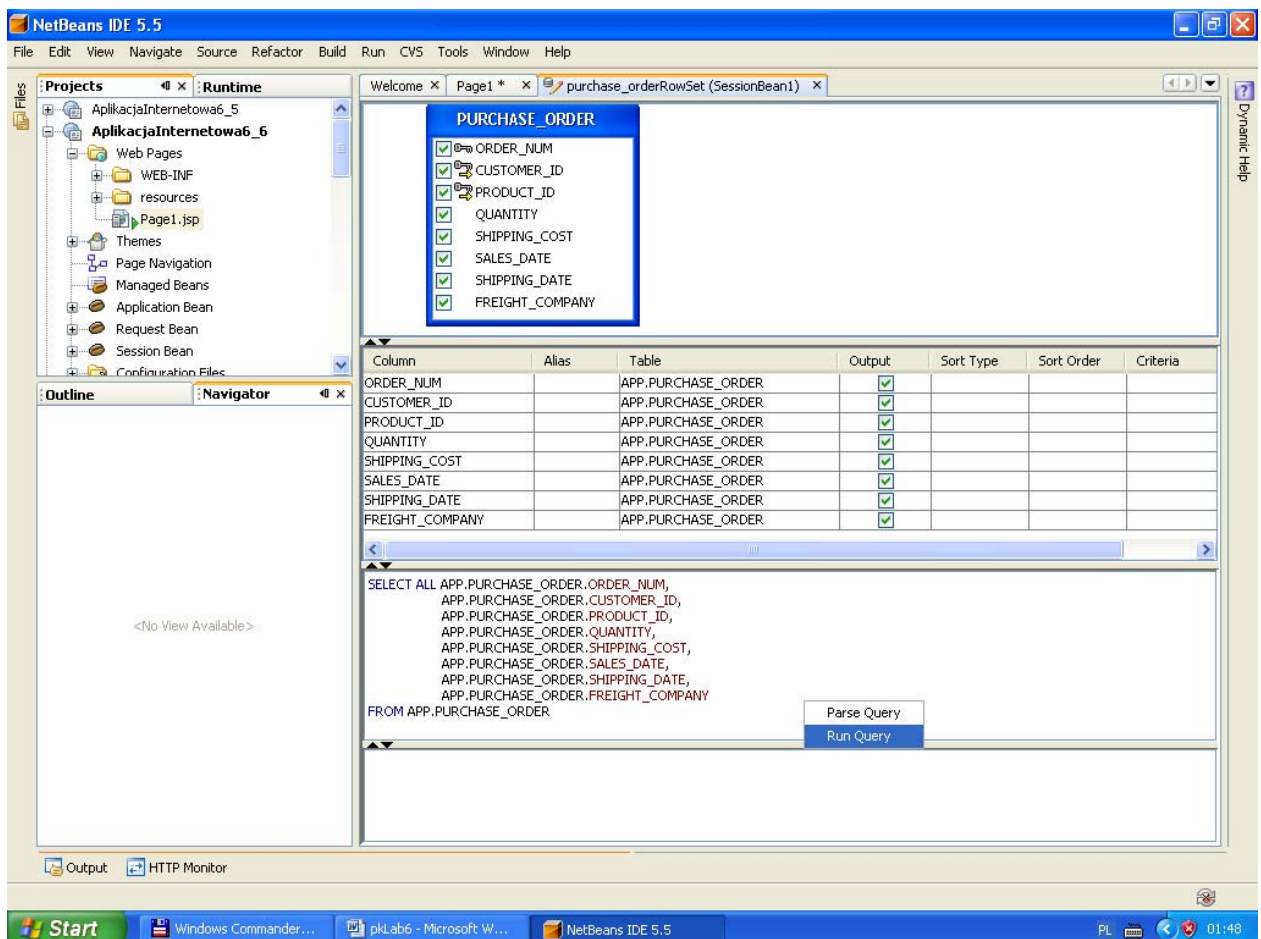
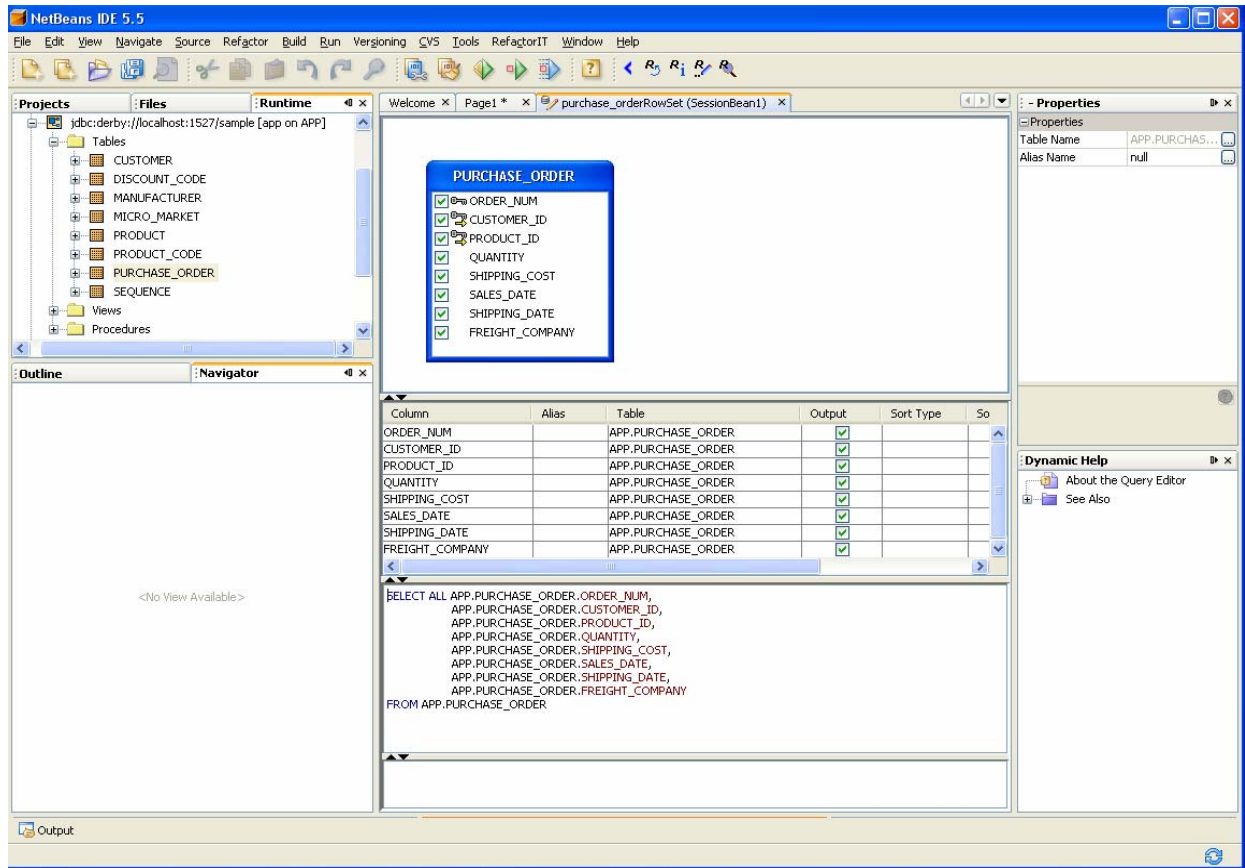
7.8. Kliknij prawym klawiszem na komponent `table1` i wybierz **Table Layout**. W ukazanym formularzu należy wybrać kolumny tabeli do wyświetlenia. Za pomocą **Ctrl+kliknięcie** wybrać następujące kolumny: `PURCHASE_ORDER.CUSTOMER_ID` i `PURCHASE_ORDER.ORDER_NUM` i nacisnąć przycisk `<` w celu usunięcia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

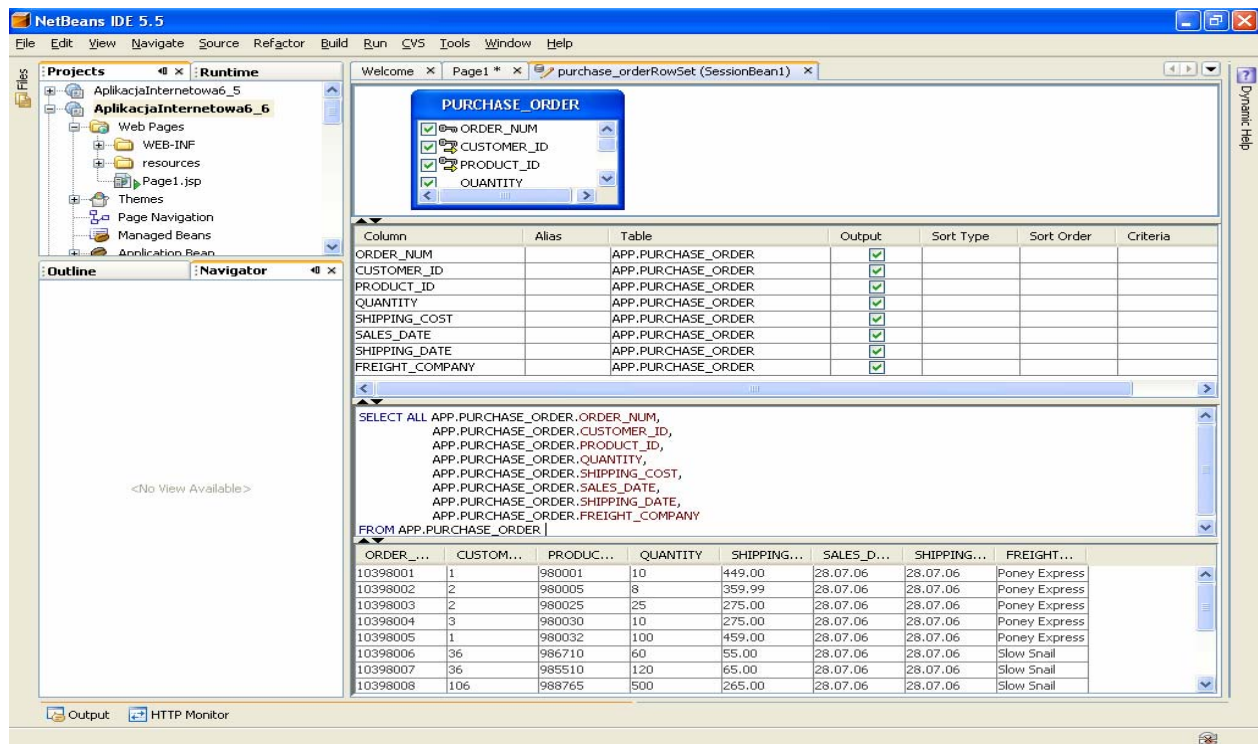




7.9. Podgląd i możliwość edycji zapytania SQL dla warstwy bazodanowej `purchase_orderRowSet` skojarzonej z `SessionBean1`. Należy wybrać w okienku **Outline** komponent `SessionBean1` i wybrać `purchase_orderRowSet` prawym klawiszem myszy. Z wyskakującego menu wybrać opcję **Edit SQL Statement**.

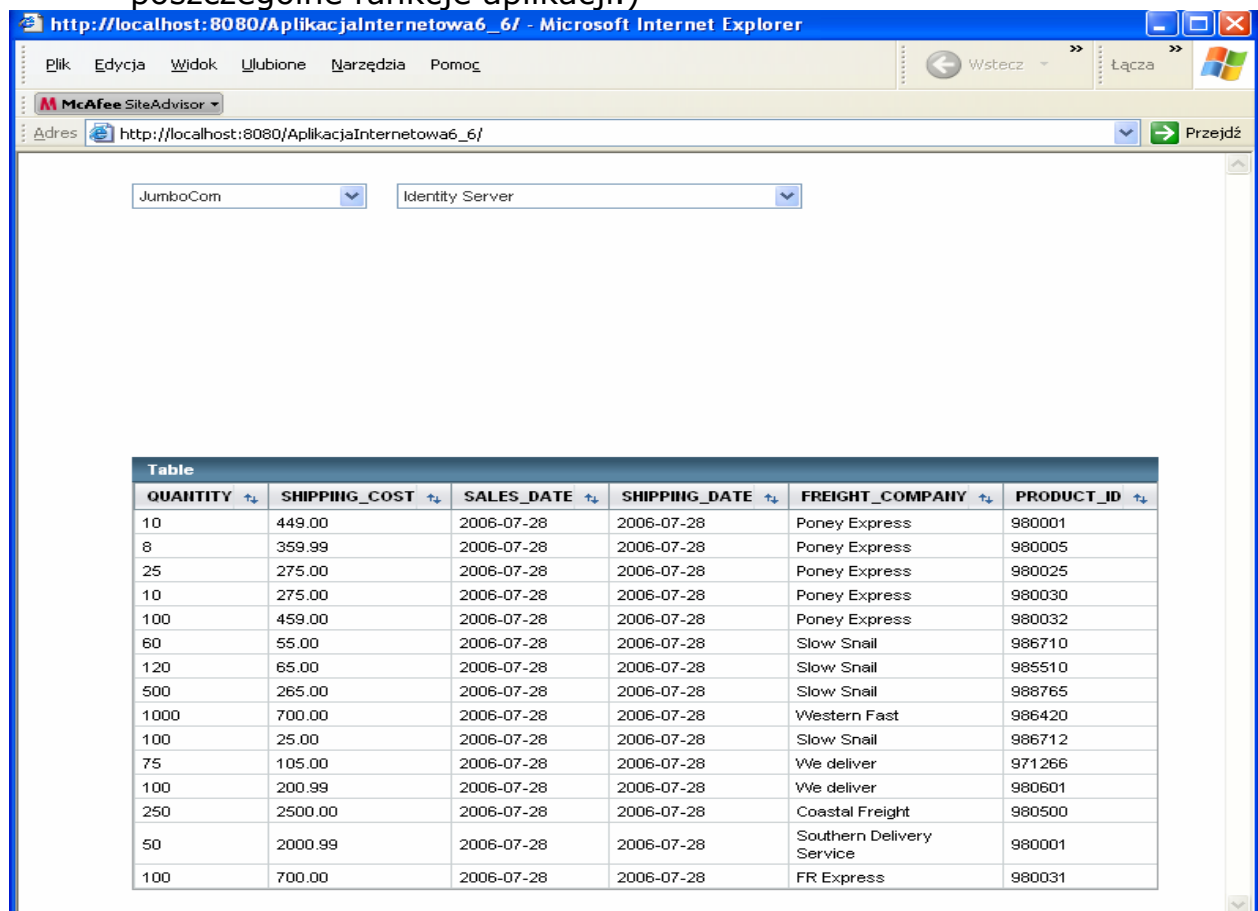






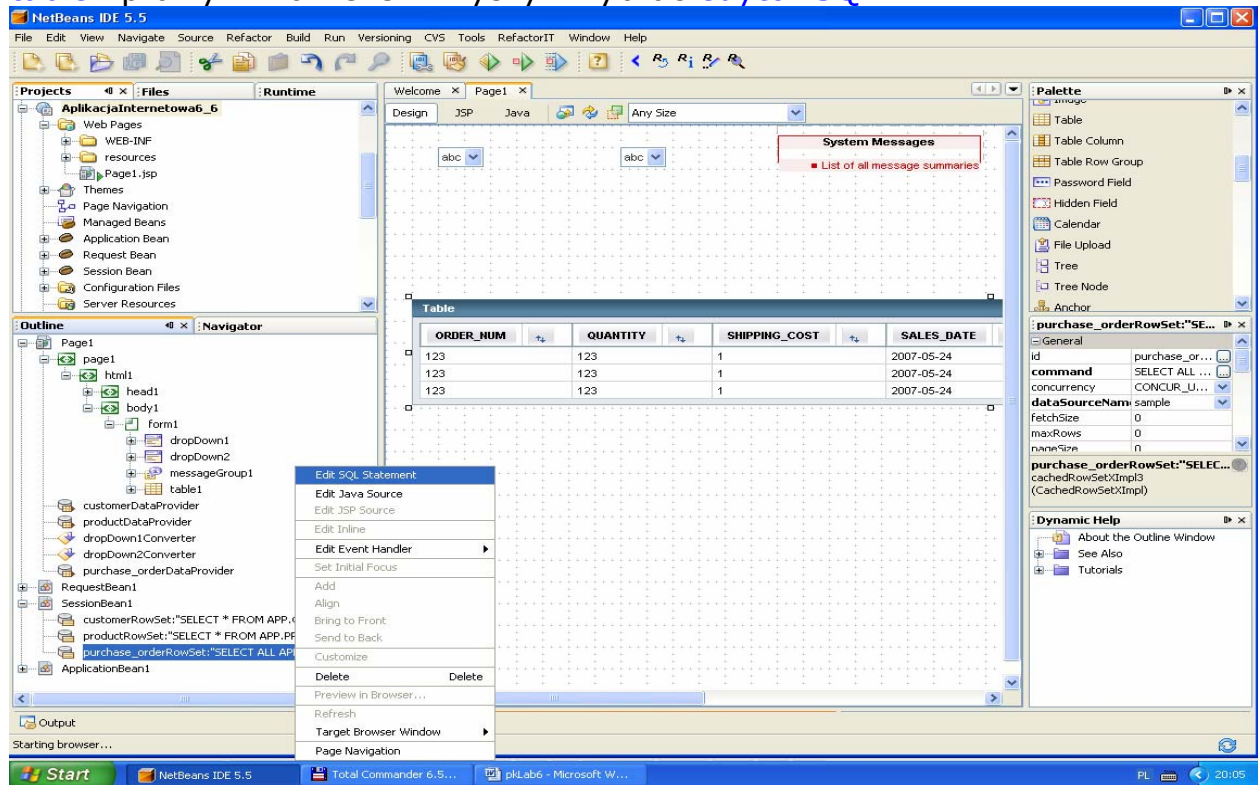
Zamknij formularz klikając na klawisz "X" zakładki `productRowSet(SssionBean1)`

7.10. 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.)

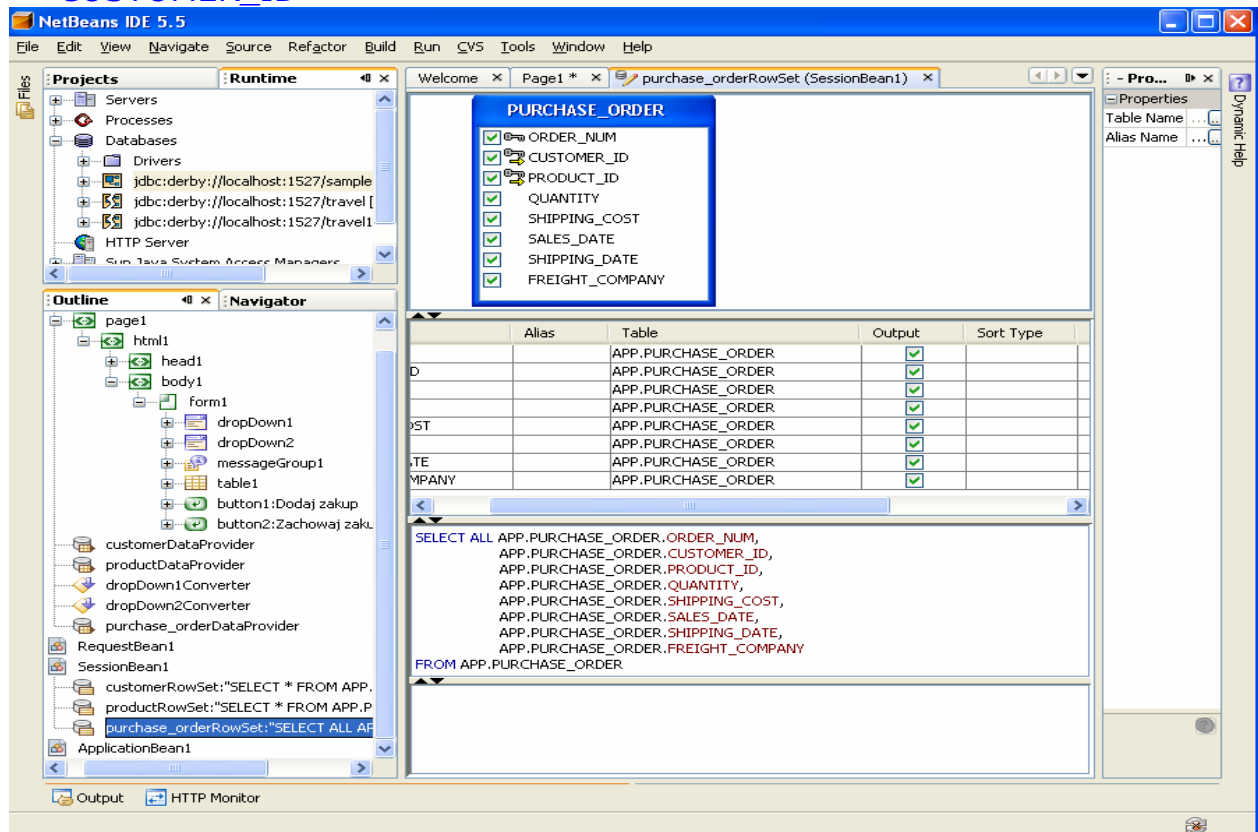


7.11. Ograniczenie wyświetlanych wierszy w tabeli do produktów firmy wybranej w komponencie `dropDown1`.

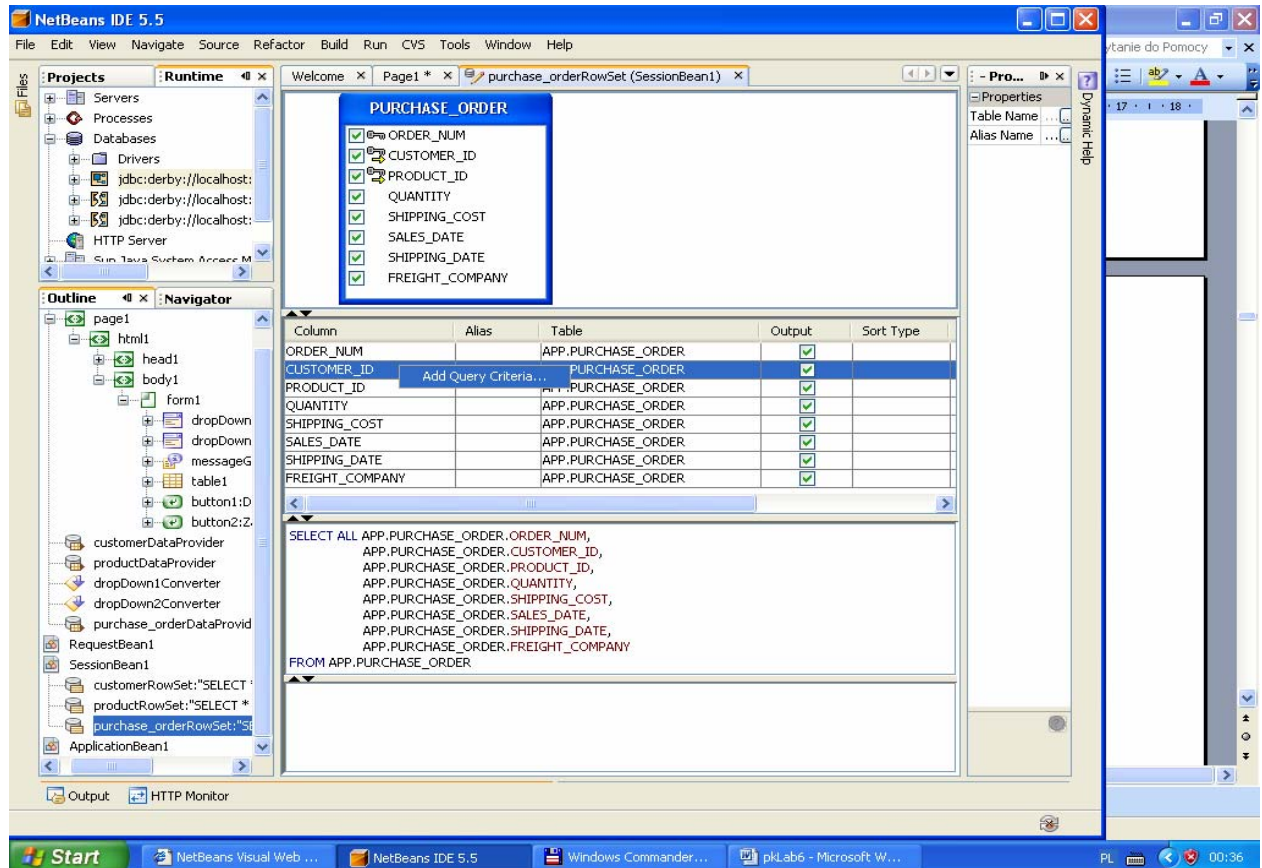
- Należy wybrać `purchase_orderRowSet` skojarzony z komponentem `table1` prawym klawiszem myszy i wybrać `edytor SQL`



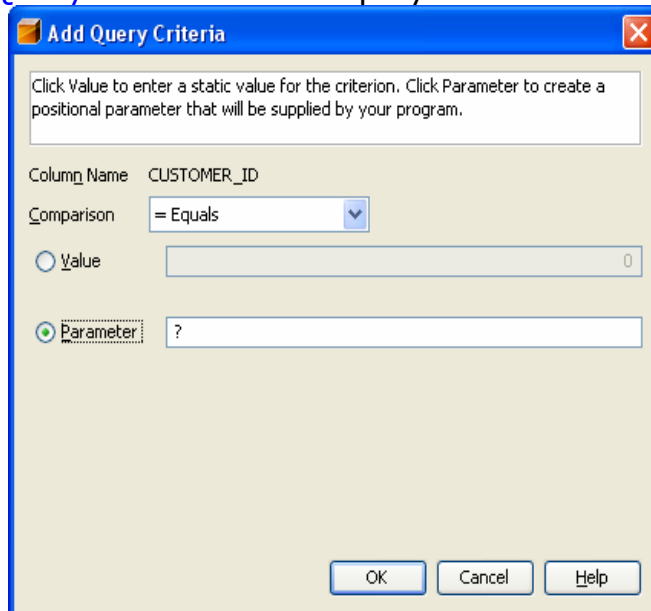
Uwaga: W tym rozwiązaniu kryterium dla zapytania `Select` jest kolumna `CUSTOMER_ID`



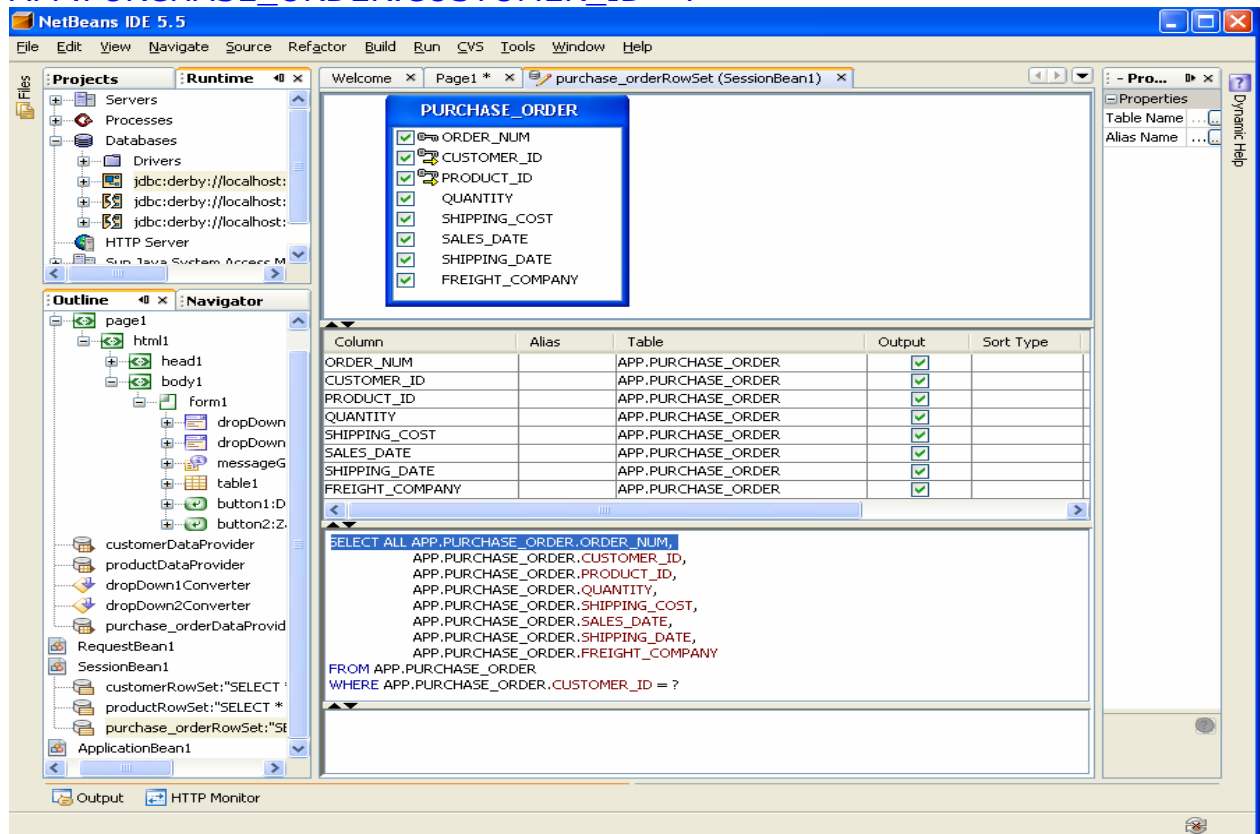
- W kolumnie **CUSTOMER_ID** kliknąć prawym klawiszem myszy i wybrać **Add Query Criteria**



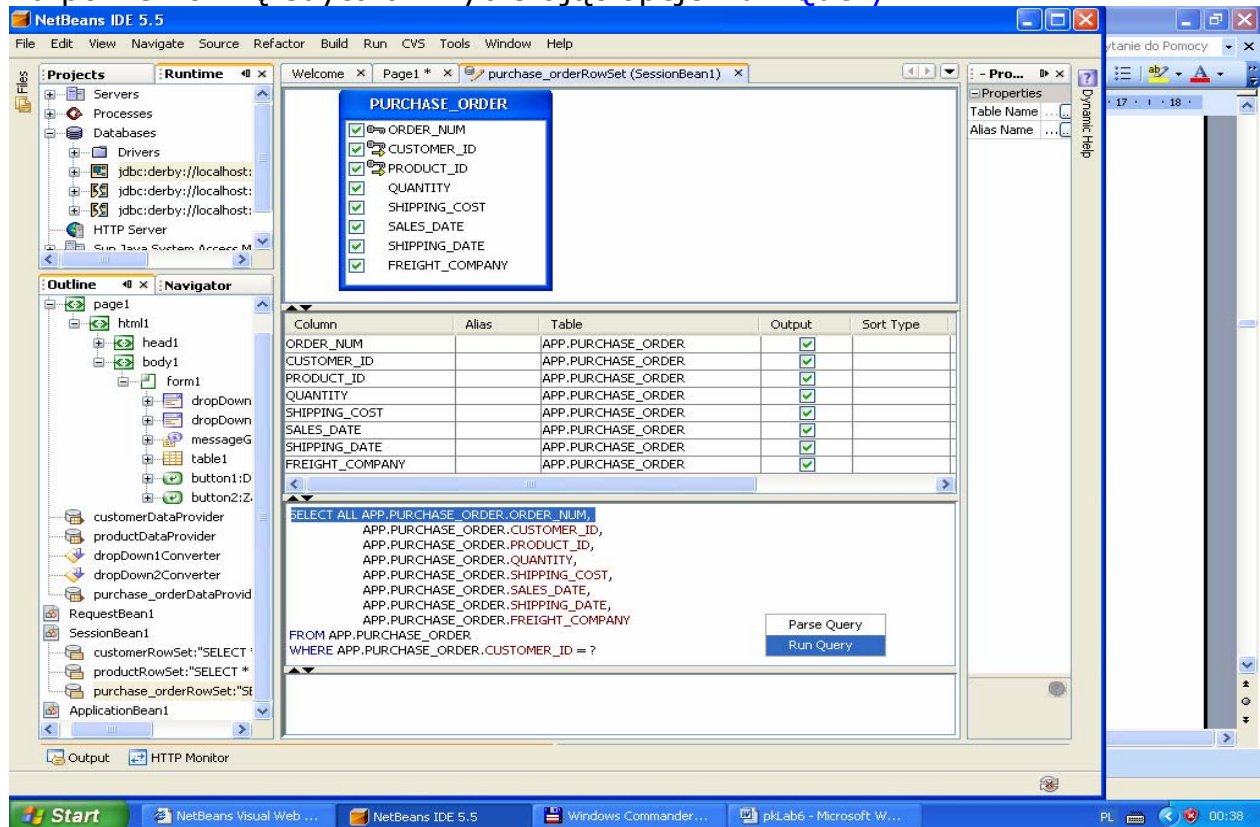
- W menu **Add Query Criteria** ustawić przycisk **Parametr**.



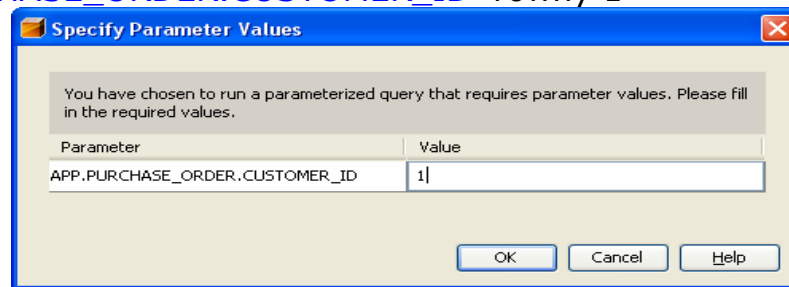
- Zapytanie zostanie zmodyfikowane o warunek **WHERE APP.PURCHASE_ORDER.CUSTOMER_ID = ?**



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



- Podanie parametru zapytania np. **APP.PURCHASE_ORDER.CUSTOMER_ID** równy 1



- Wynik zapytania

The screenshot shows the NetBeans IDE 5.5 interface. The 'Dynamic Help' window is open, displaying the results of a query. The table below shows the columns and their corresponding values from the database.

Column	Alias	Table	Output	Sort Type	Sort Order
ORDER_NUM		APP.PURCHASE_ORDER	<input checked="" type="checkbox"/>		
CUSTOMER_ID		APP.PURCHASE_ORDER	<input checked="" type="checkbox"/>		
PRODUCT_ID		APP.PURCHASE_ORDER	<input checked="" type="checkbox"/>		
QUANTITY		APP.PURCHASE_ORDER	<input checked="" type="checkbox"/>		
SHIPPING_COST		APP.PURCHASE_ORDER	<input checked="" type="checkbox"/>		

The SQL query displayed is:

```
SELECT ALL APP.PURCHASE_ORDER.ORDER_NUM,
APP.PURCHASE_ORDER.CUSTOMER_ID,
APP.PURCHASE_ORDER.PRODUCT_ID,
APP.PURCHASE_ORDER.QUANTITY,
APP.PURCHASE_ORDER.SHIPPING_COST,
APP.PURCHASE_ORDER.SALES_DATE,
APP.PURCHASE_ORDER.SHIPPING_DATE,
APP.PURCHASE_ORDER.FREIGHT_COMPANY
FROM APP.PURCHASE_ORDER
WHERE APP.PURCHASE_ORDER.CUSTOMER_ID = ?
```

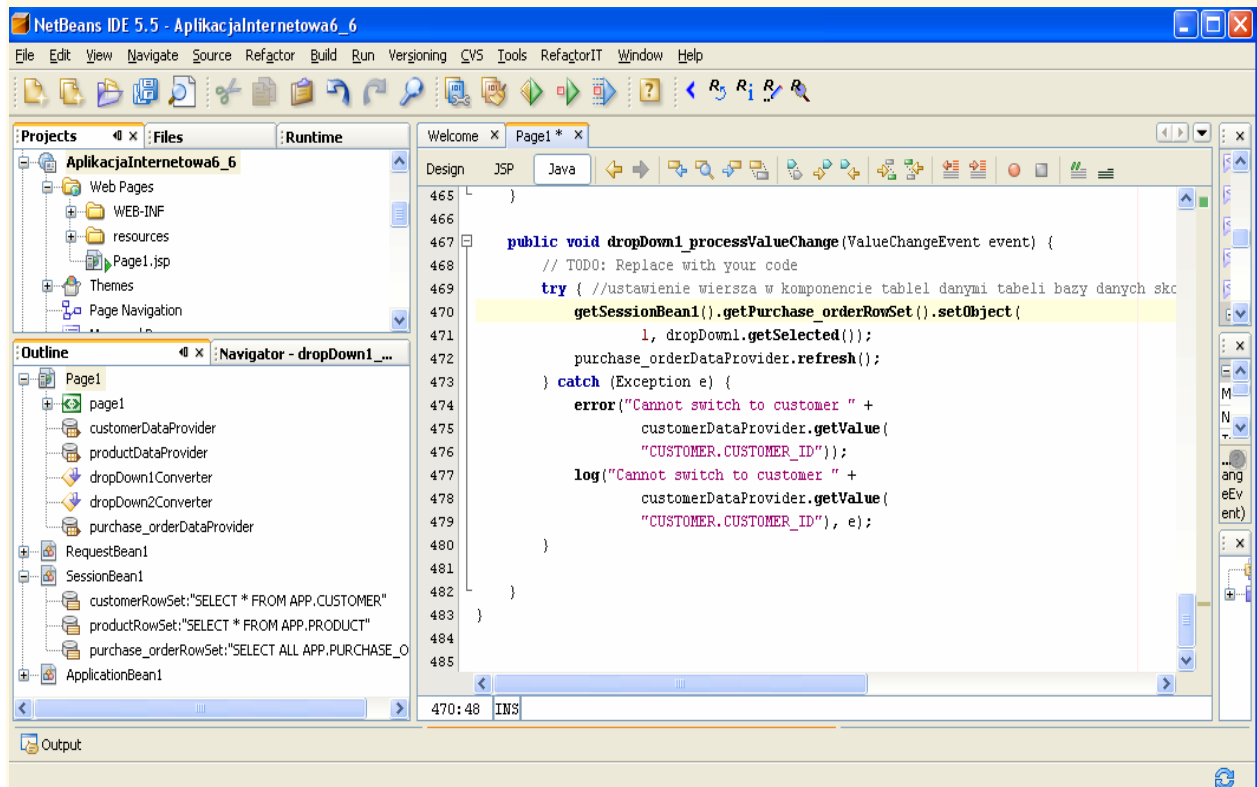
The data grid below the query shows the following results:

ORDER_...	CUSTOM...	PRODUC...	QUANTITY	SHIPPING...	SALES_D...	SHIPPING...	FREIGHT...
10398001	1	980001	10	449.00	28.07.06	28.07.06	Poney Express
10398005	1	980032	100	459.00	28.07.06	28.07.06	Poney Express

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

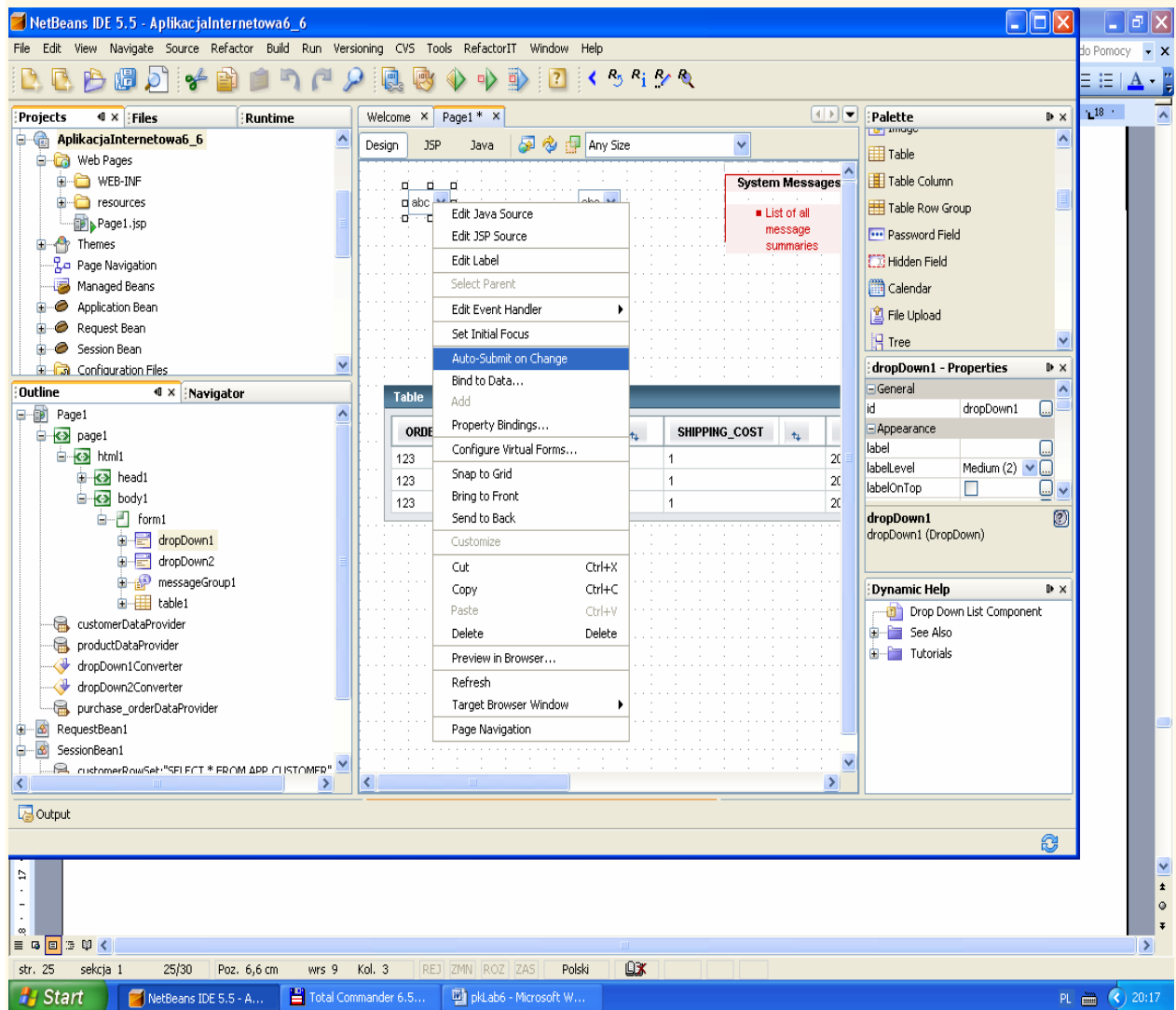
8.1. Należy kliknąć podwójnie na kontrolkę [dropDown1](#) i obsłużyć zdarzenie [ProcessValueChange](#) za pomocą metody [dropDown1_processValueChange](#)

```
public void dropDown1_processValueChange(ValueChangeEvent event)
{
    try { //ustawienie wiersza w komponentcie table1 danymi tabeli bazy danych skojarzonej z komponentem dropDown1 i
//wybranymi w tym komponentcie
        getSessionBean1().getPurchase_orderRowSet().setObject( 1,
            dropDown1.getSelected());
        purchase_orderDataProvider.refresh();
    } catch (Exception e) {
        error("Cannot switch to customer " +
            customerDataProvider.getValue(
                "CUSTOMER.CUSTOMER_ID"));
        log("Cannot switch to customer " +
            customerDataProvider.getValue(
                "CUSTOMER.CUSTOMER_ID"), e);
    }
}
```



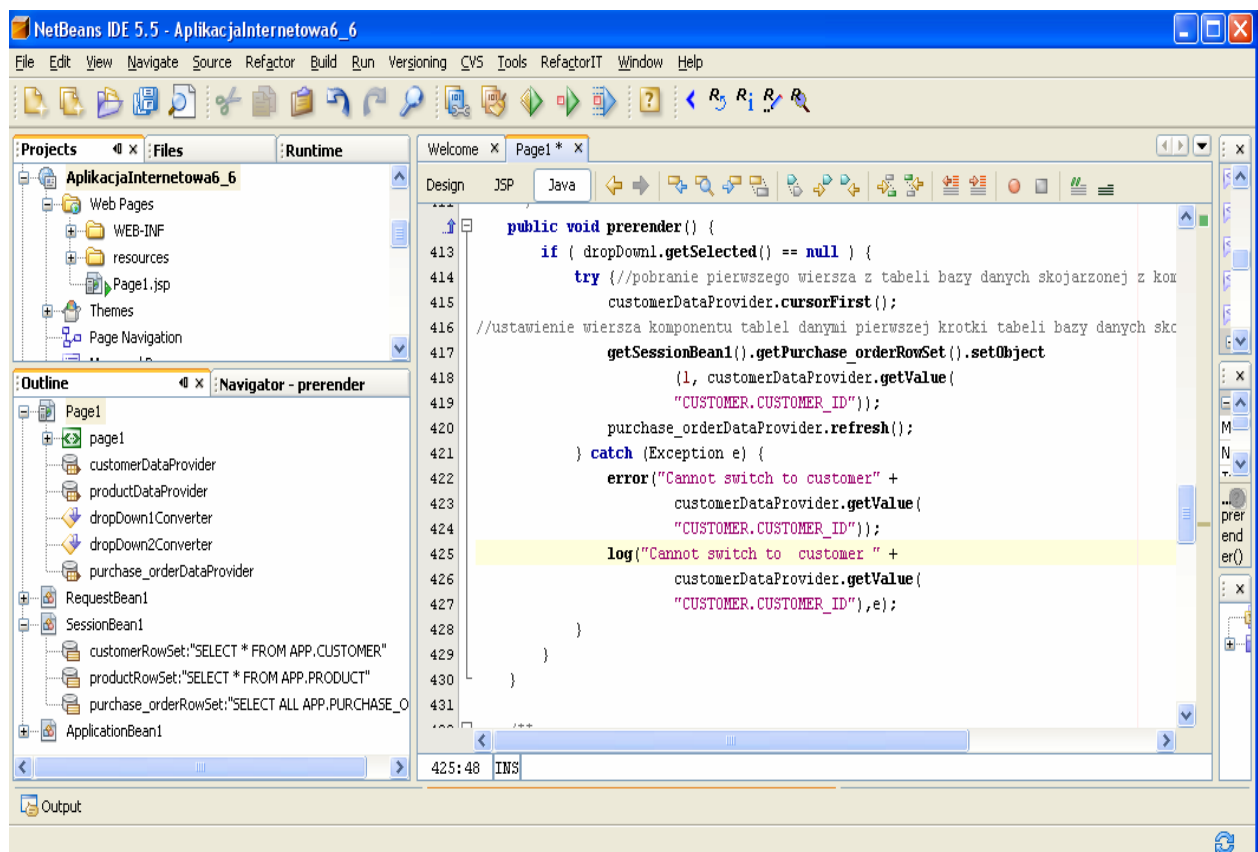
8.2. Przejdź do trybu **Design** strony **Page1**, zaznacz komponent **dropDown1** i prawym klawiszem myszy kliknij nad tym zaznaczonym komponentem. Z wyskakującego menu ustaw **Auto-Submit on Change**.

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

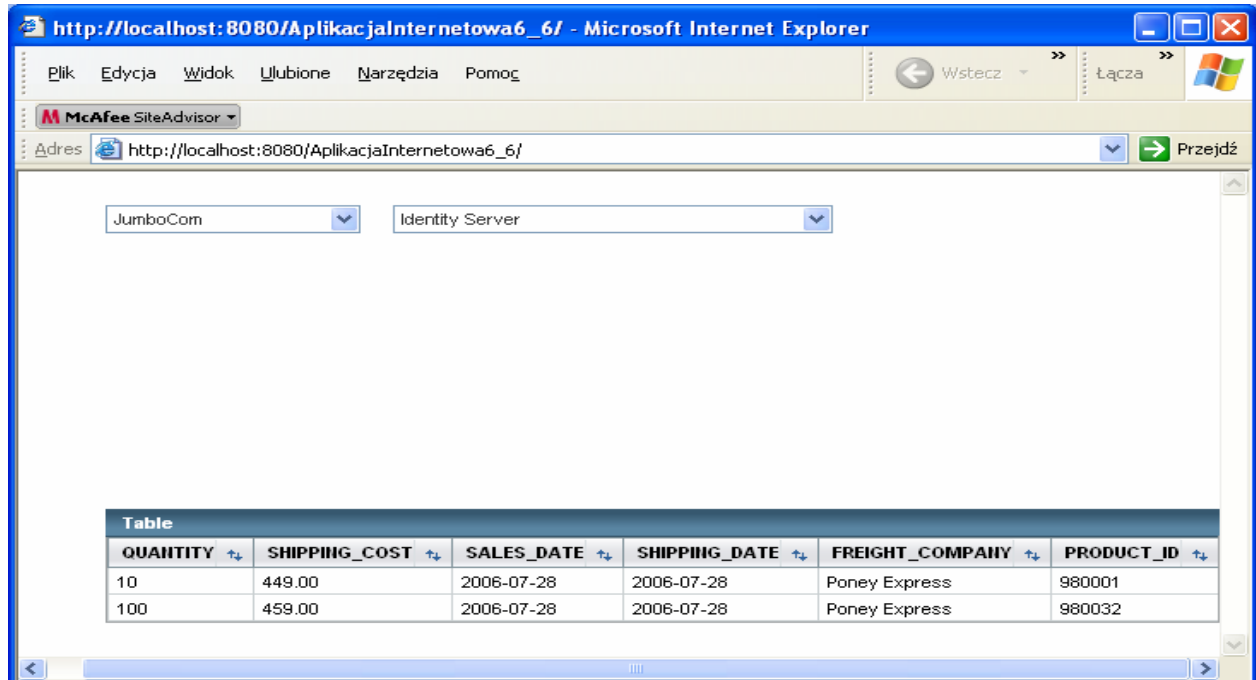


8.3. metoda prerender() //inicjowanie zawartości komponentu table1

```
public void prerender()
{
    if ( dropDown1.getSelected() == null ) {
        try { //pobranie pierwszego wiersza z tabeli bazy danych skojarzonej z komponentem dropDown1
            customerDataProvider.cursorFirst();
            //ustawienie wiersza komponentu table1 danymi pierwszej krotki tabeli bazy danych skojarzonej z komponentem dropDown1
            getSessionBean1().getPurchase_orderRowSet().setObject
                (1, customerDataProvider.getValue(
                    "CUSTOMER.CUSTOMER_ID"));
            purchase_orderDataProvider.refresh();
        } catch (Exception e) {
            error("Cannot switch to manufacturer" +
                customerDataProvider.getValue(
                    "CUSTOMER.CUSTOMER_ID"));
            log("Cannot switch to manufacturer " +
                customerDataProvider.getValue(
                    "CUSTOMER.CUSTOMER_ID"),e);
        }
    }
}
```

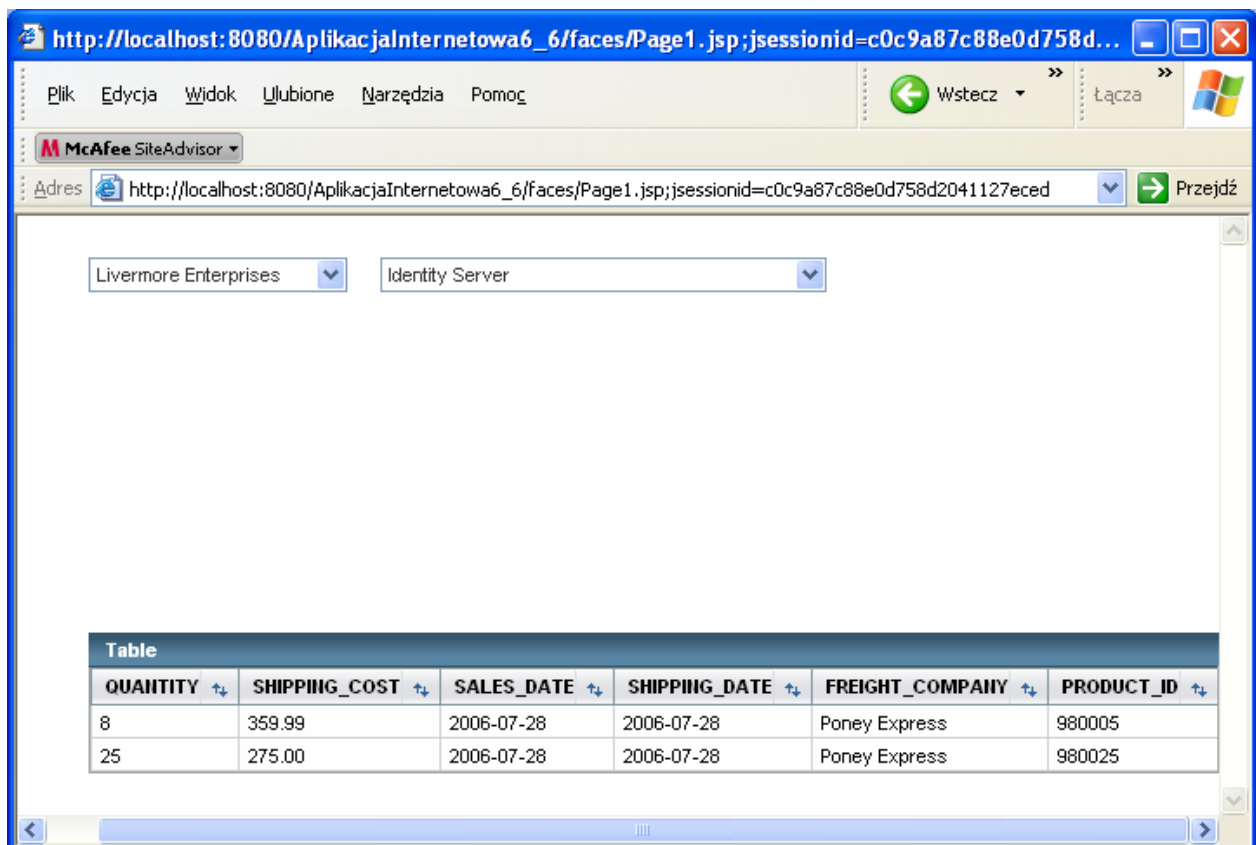


8.3. Uruchom aplikację (Kliknij prawym klawiszem myszy w oknie **Project** na nazwę projektu, w ukazanym oknie uruchom kolejno **Build Project**, **Deploy Project**, **Run Project** lub tylko **Run Project**) i uruchom poszczególne funkcje aplikacji.



Table

QUANTITY	SHIPPING_COST	SALES_DATE	SHIPPING_DATE	FREIGHT_COMPANY	PRODUCT_ID
10	449.00	2006-07-28	2006-07-28	Poney Express	980001
100	459.00	2006-07-28	2006-07-28	Poney Express	980032

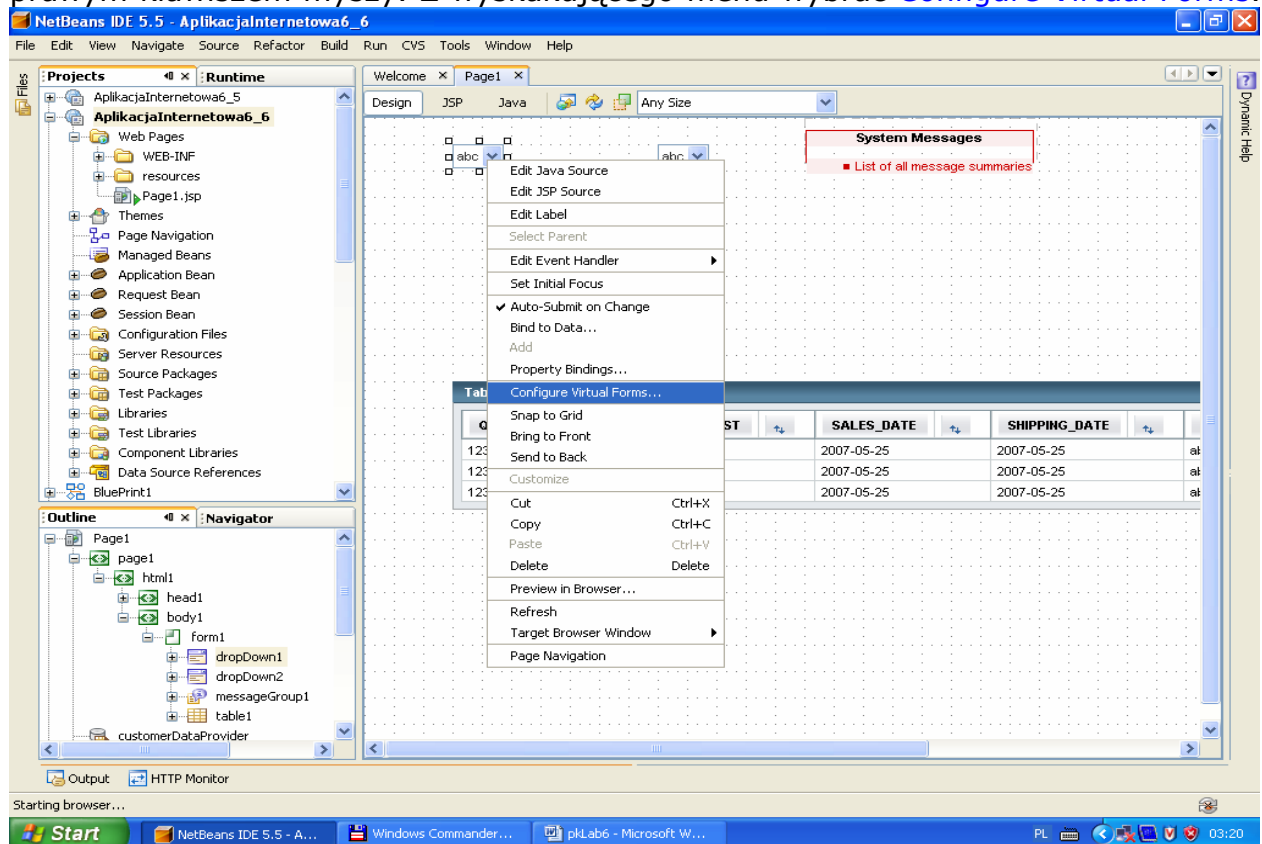


Table

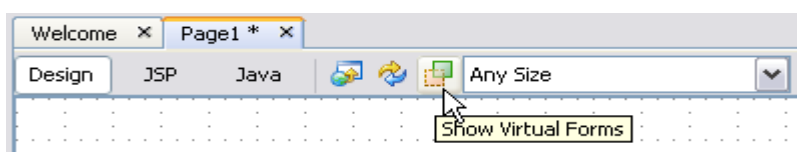
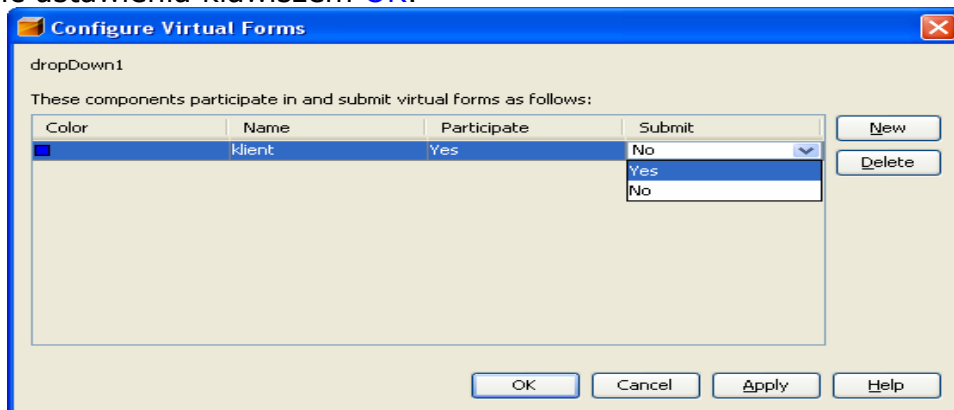
QUANTITY	SHIPPING_COST	SALES_DATE	SHIPPING_DATE	FREIGHT_COMPANY	PRODUCT_ID
8	359.99	2006-07-28	2006-07-28	Poney Express	980005
25	275.00	2006-07-28	2006-07-28	Poney Express	980025

9. Konfiguracja wirtualnego formularza (Configure Virtual Forms) (laboratorium 4)

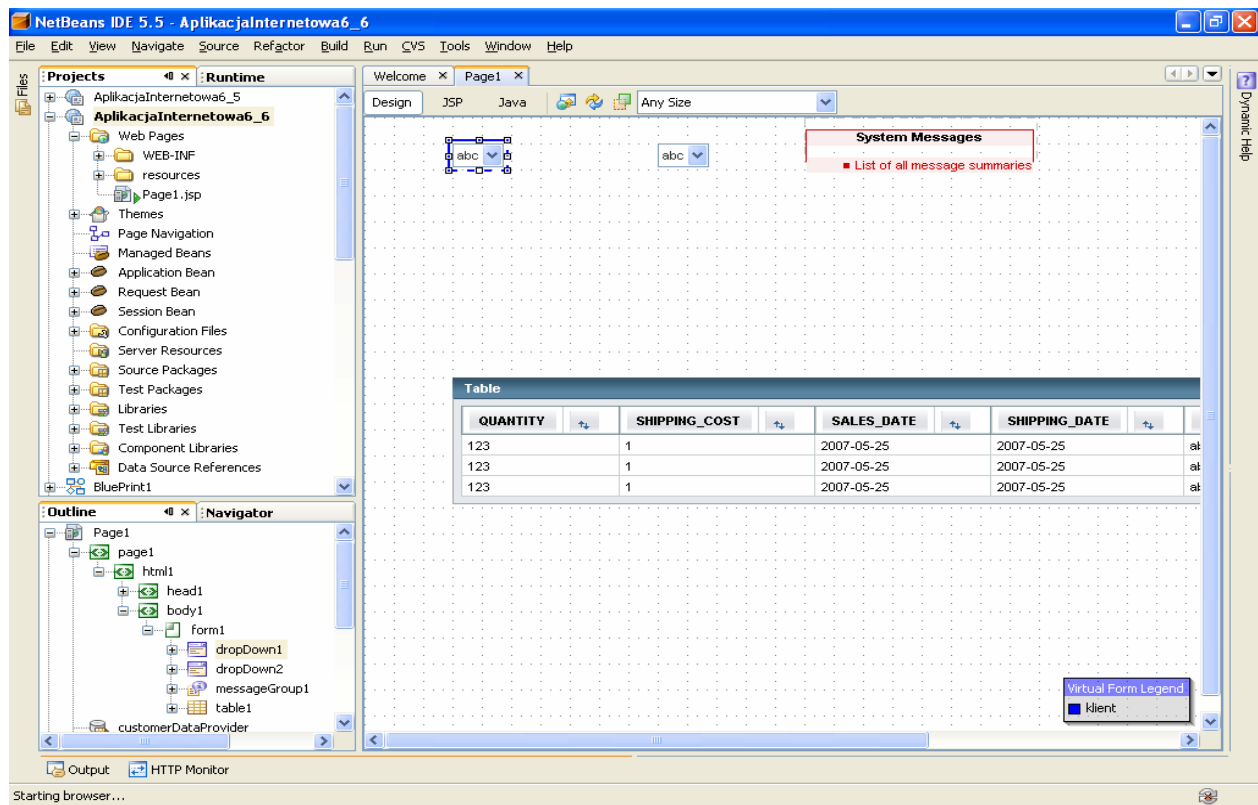
9.1. W trybie Design strony Page1 należy wybrać komponent dropDown1 i kliknąć prawym klawiszem myszy. Z wyskakującego menu wybrać Configure Virtual Forms.



9.2. W formularzu opcji nacisnąć klawisz New. Nowy formularz nazwać np. klient i oraz kliknąć dwa razy w kolumny Participate i Submit i w obu ustawić Yes. Zatwierdzić ustawienia klawiszem OK.

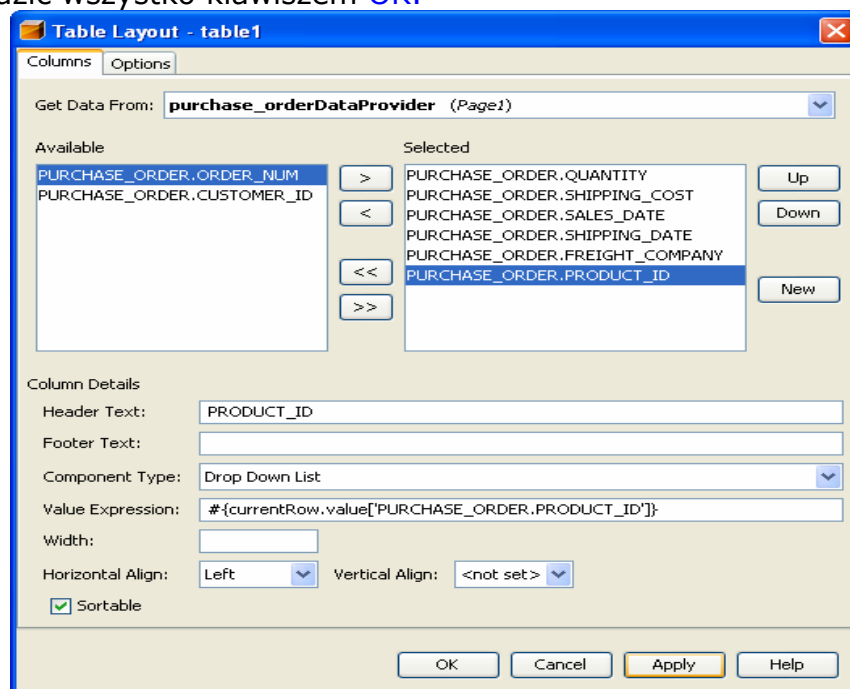


W wyniku ustawienia wirtualnego formularza nie są sprawdzane dane wyświetlane w komponencie table1. W trybie Design komponent należący do wirtualnego formularza jest podświetlony, pod warunkiem, że zostanie on pokazany po kliknięciu na ikonę pokazaną powyżej na belce narzędziowej.

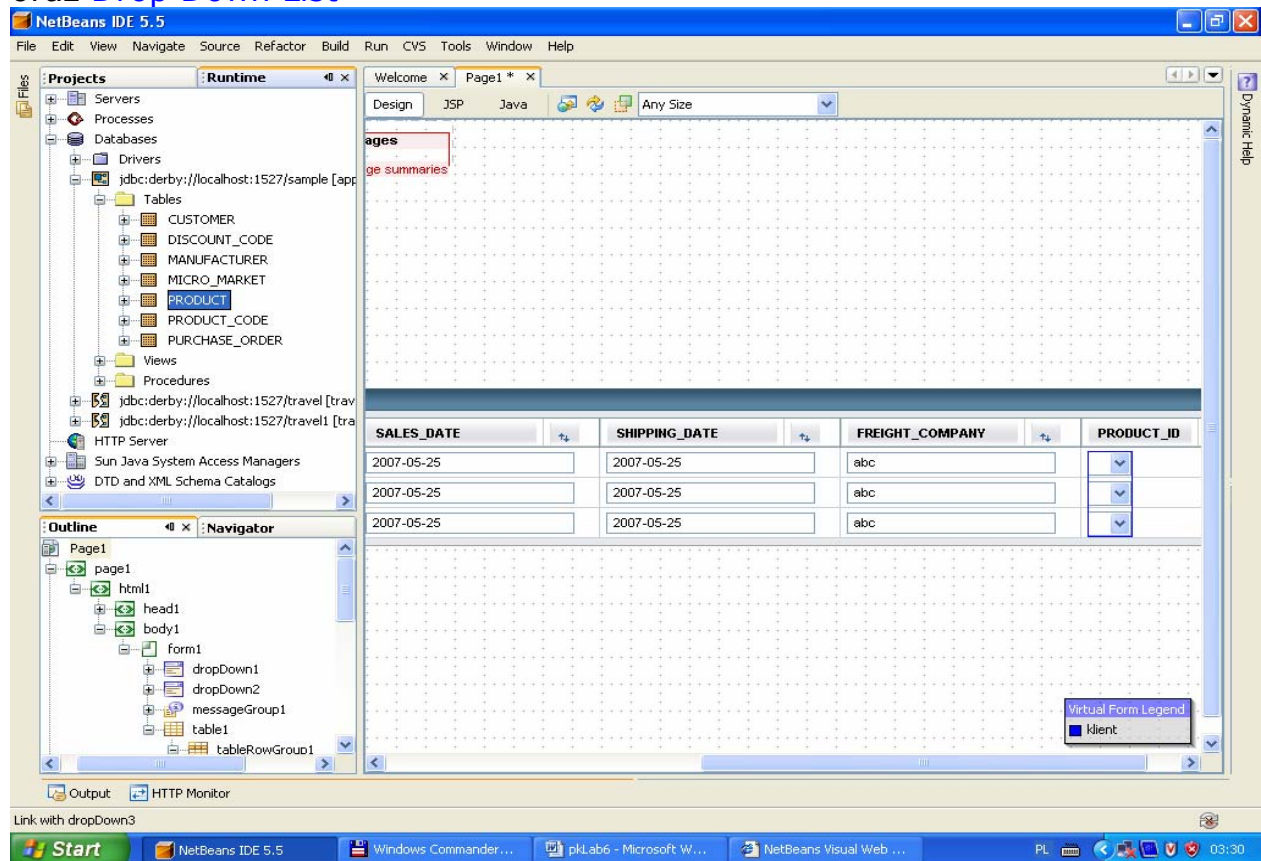


9.3. Wstawianie komponentów wejściowych do komórek komponentu `table1`

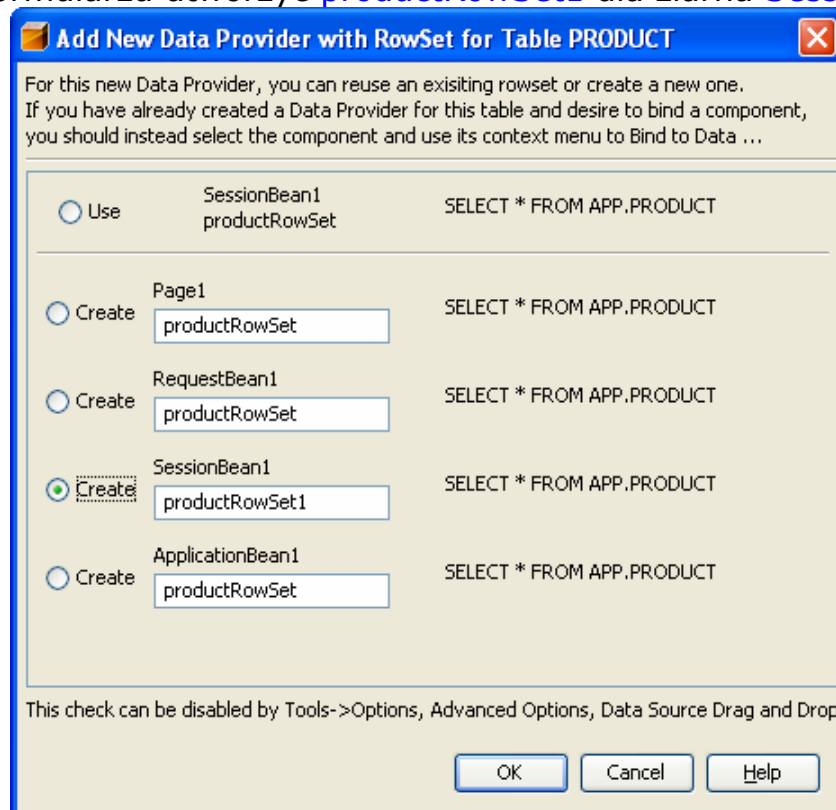
- Po zaznaczeniu komponentu `table1` należy kliknąć prawym klawiszem myszy i wybrać z wyskakującego menu wybrać `Table Layout`.
- W ukazanym formularzu wybrać kolejno kolumny tabeli w prawej części formularza i ustawić typ komponentu `Text Field` w liście `Komponent Type` i zatwierdzić kolejne ustawienia klawiszem `Apply`. Dla kolumny `PURCHASE_ORDER.PRODUCT_ID` należy wstawić komponent `Drop Down List`. Zatwierdzić wszystko klawiszem `OK`.



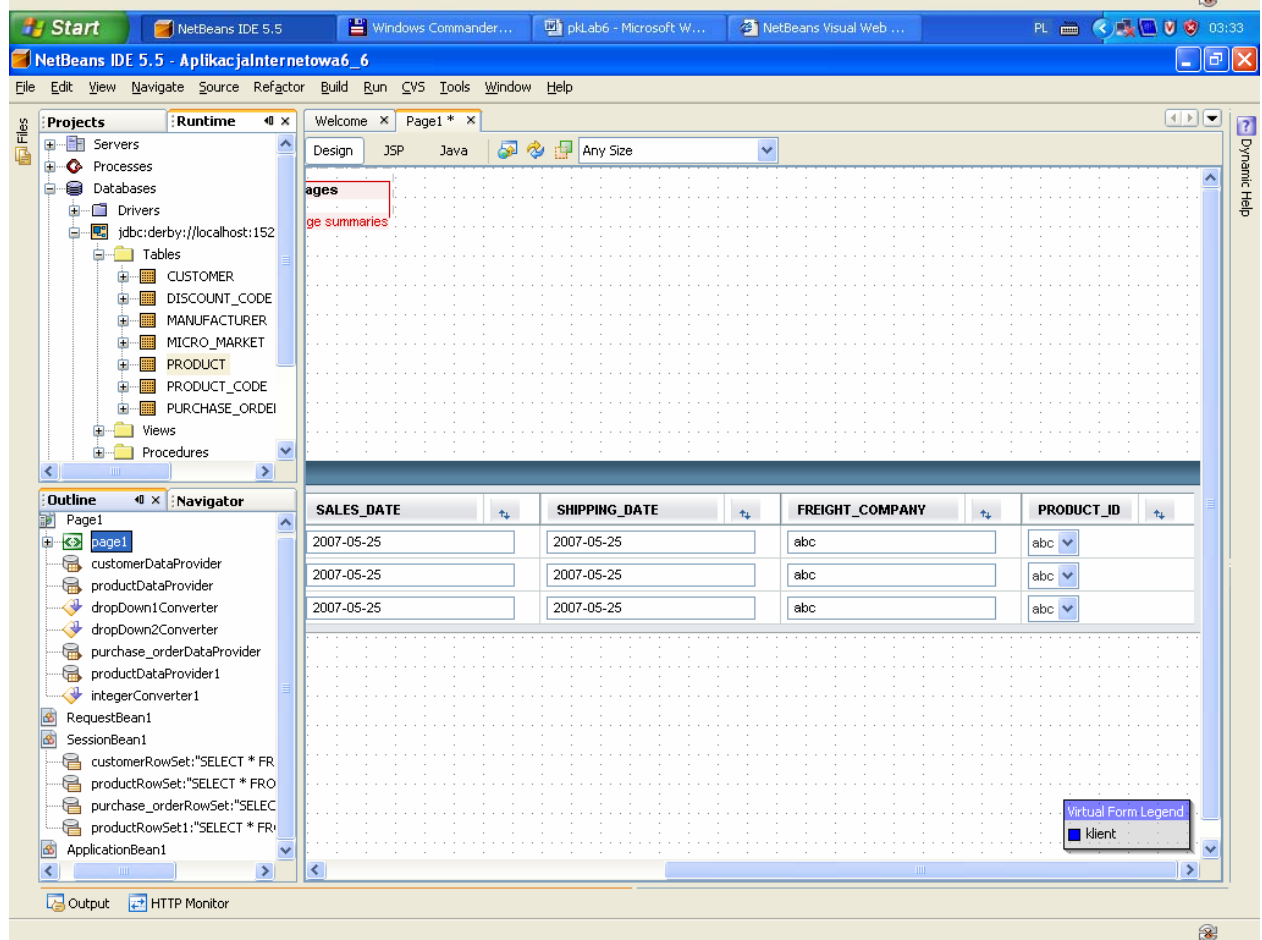
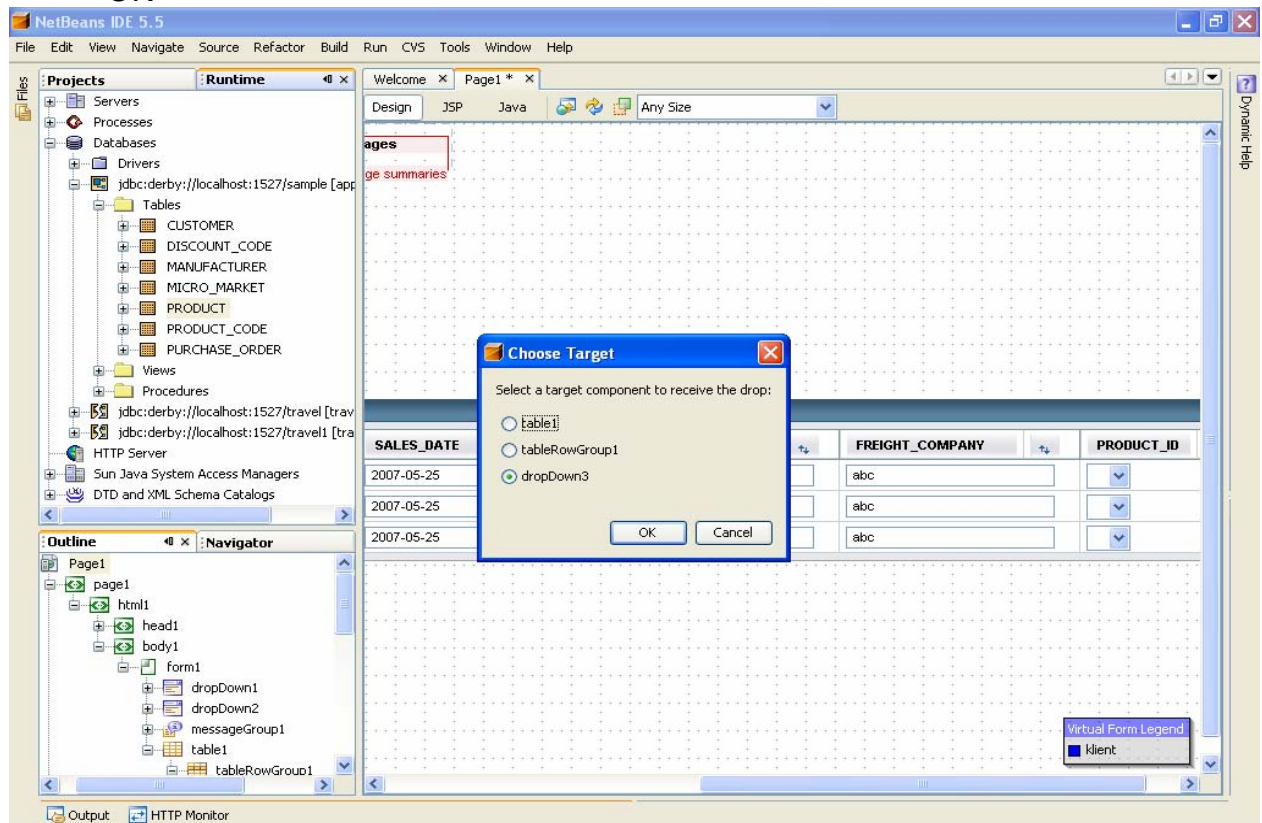
- W trybie Design wiersze tabeli zawierają komponenty typu Text Field oraz Drop Down List



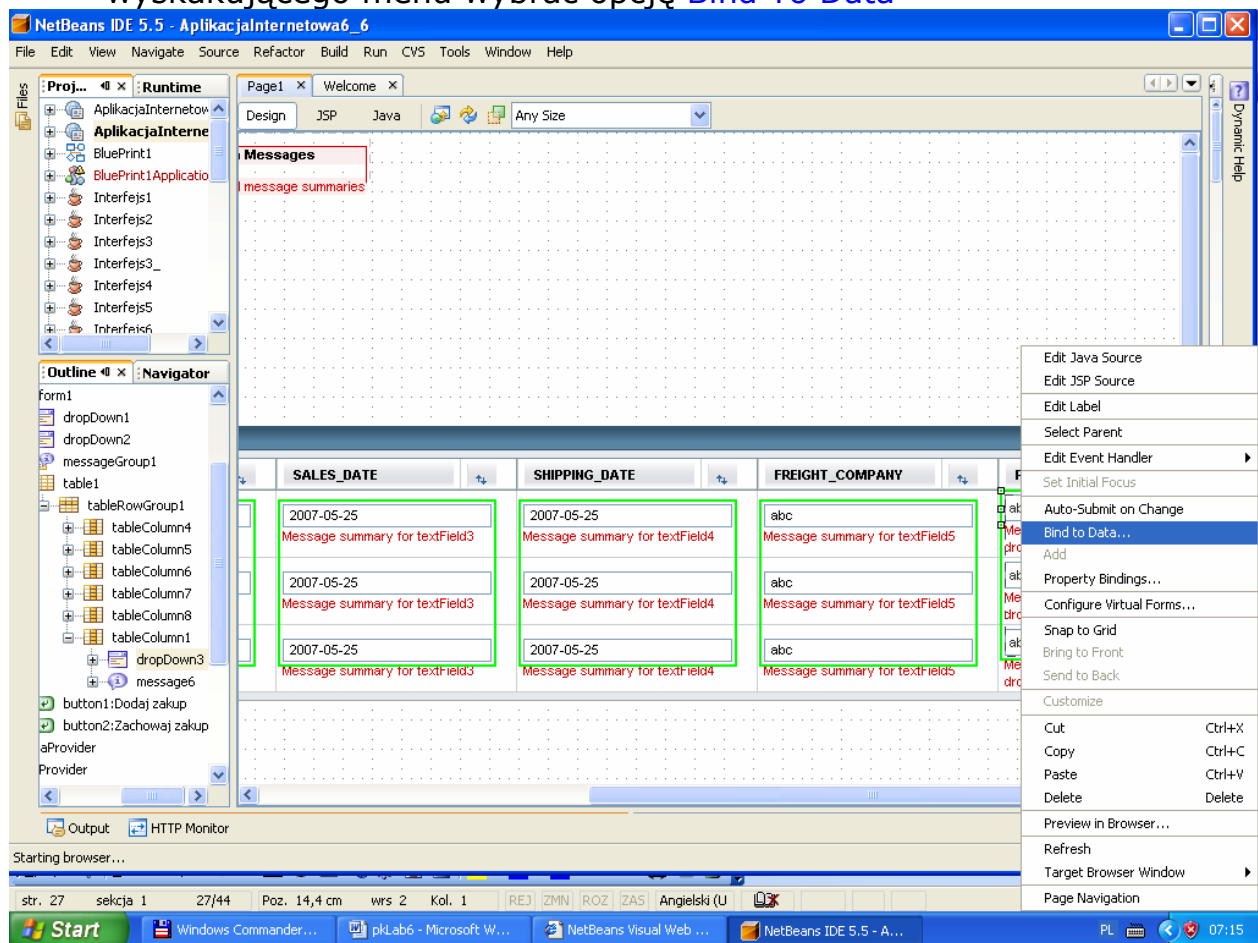
- Na nowy komponent dropDown3 przeciągnąć tabelę PRODUCT. W ukazanym formularzu utworzyć productRowSet1 dla ziarna SessionBean1.



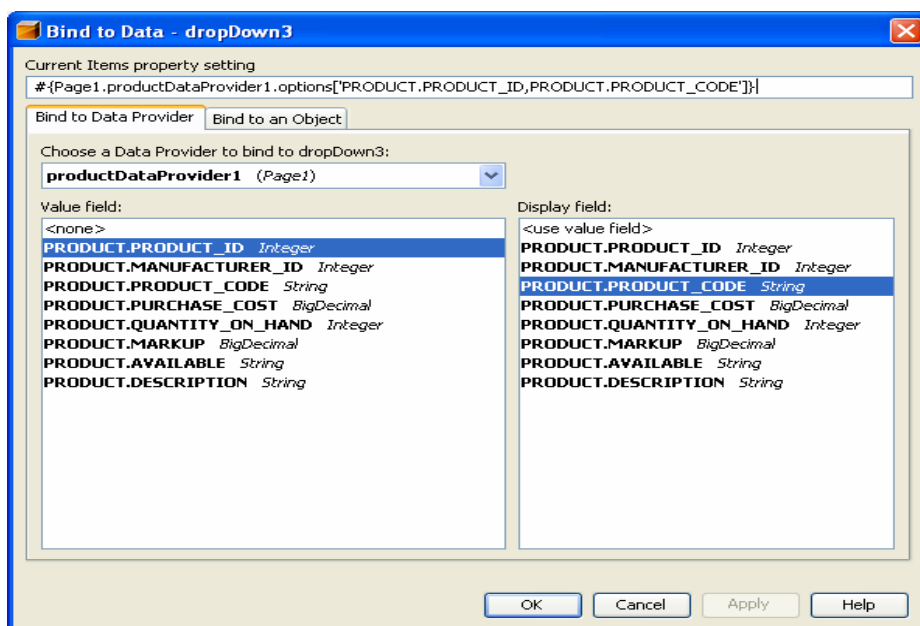
- W okienku **Choose Target** wybrać **dropDown3** i zatwierdzić klawiszem **OK**



- Zaznaczyć komponent **dropDown3** i prawym klawiszem wybrać z wyskakującego menu wybrać opcję **Bind To Data**

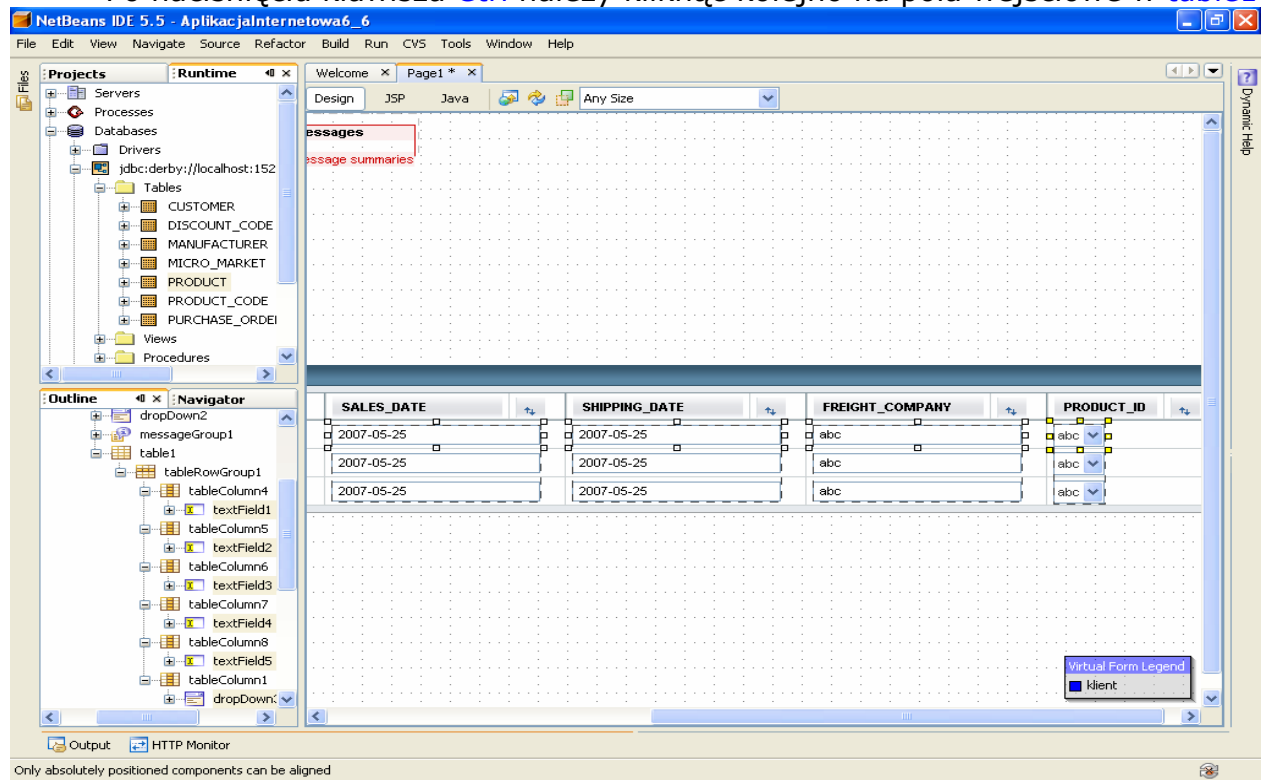


- W wywołanym formularzu wybrać zakładki **Value** oraz **Field** i zaznaczyć **PRODUCT.PRODUCT_ID** jako domyślna wartość w programie oraz w zakładce **Display** wybrać **PRODUCT.PRODUCT_CODE** jako kolumnę do wyświetlania na pozycjach komponentu **dropDown3**.

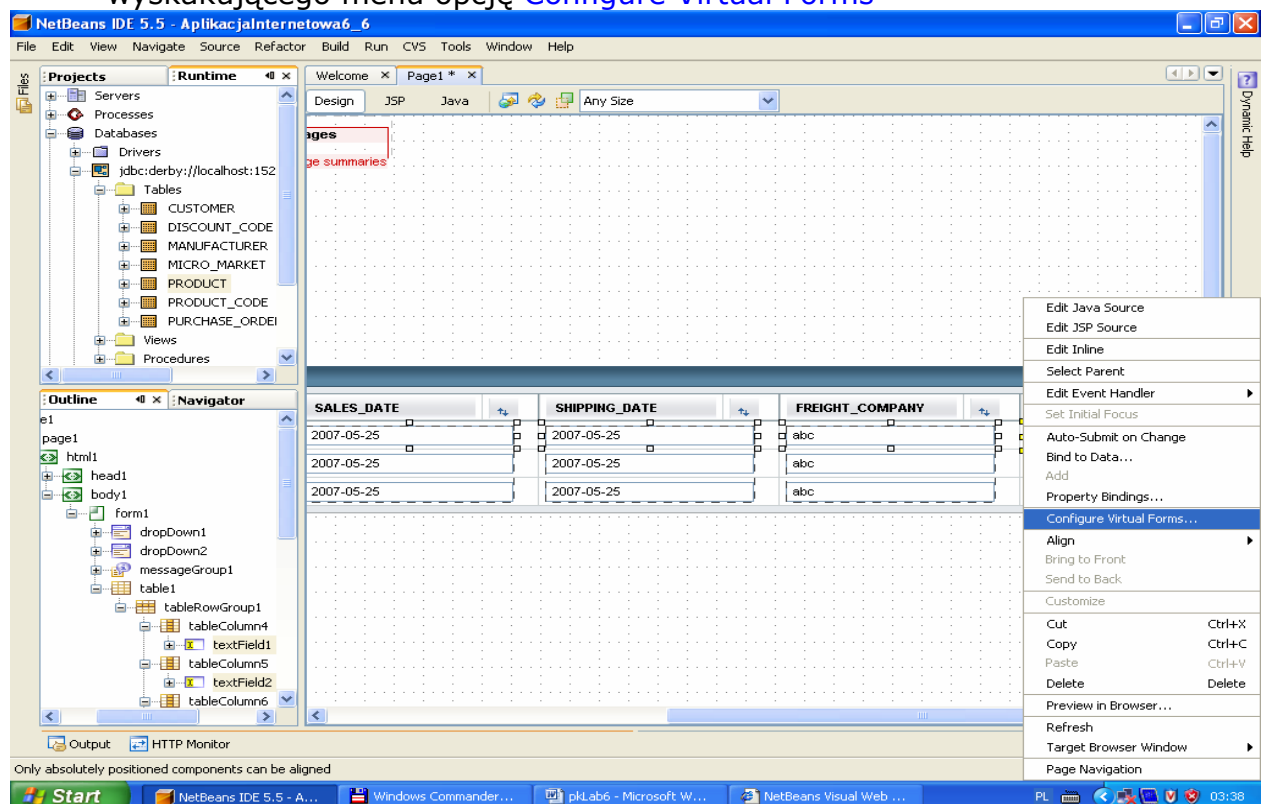


9.4. Utworzenie wirtualnego formularza dla pól wejściowych w komponencie `table1`

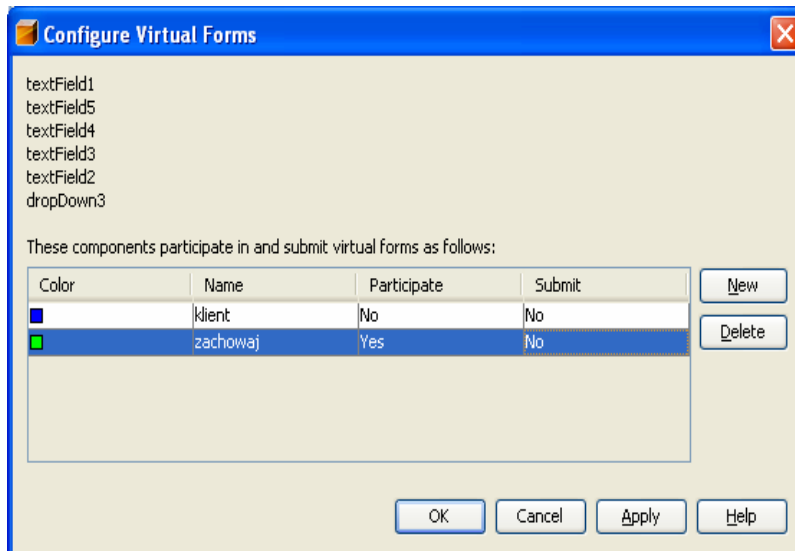
- Po naciśnięciu klawisza **Ctrl** należy kliknąć kolejno na pola wejściowe w `table1`



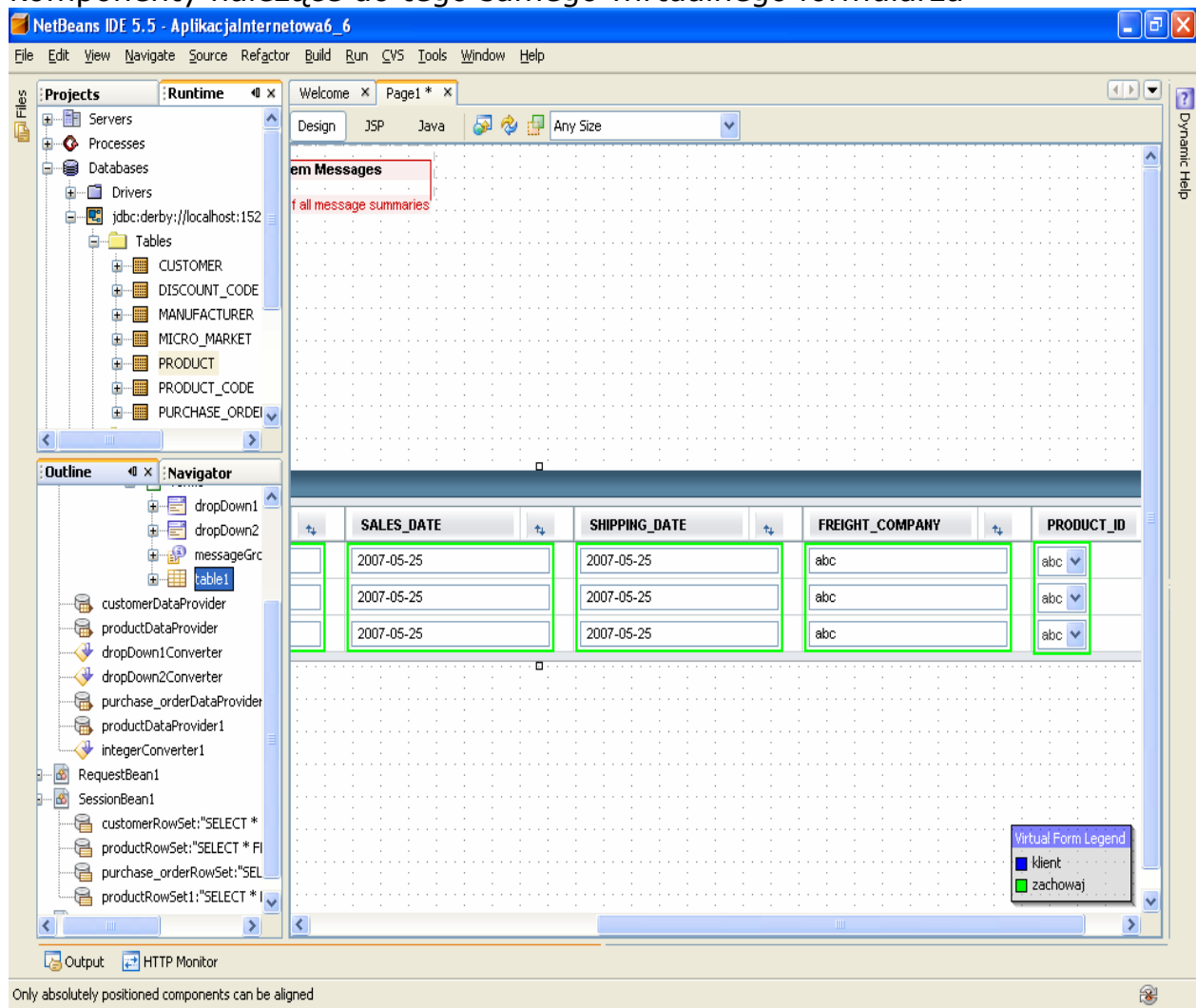
- Następnie po naciśnięciu prawego klawisza myszy należy wybrać z wyskakującego menu opcję **Configure Virtual Forms**

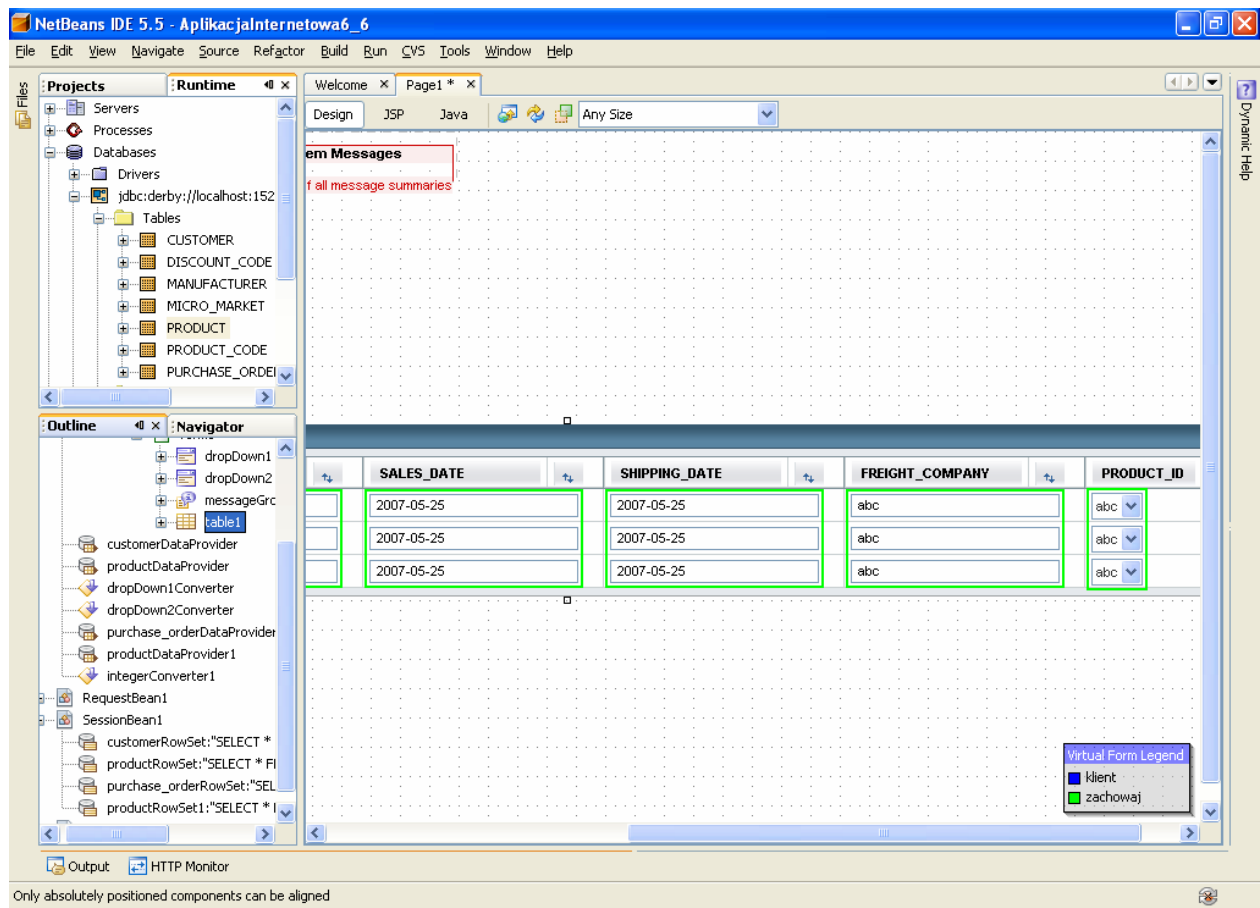


- Po naciśnięciu klawisza **New** należy wstawić nowy formularz np. o nazwie `zachowaj`. Należy podwójnie kliknąć dwukrotnie kolumnę `Participate` i ustawić `Yes` i zatwierdzić nowy formularz klawiszem **OK**.



- Na stronie **Page1** w trybie **Design** zaznaczone są odrębnymi kolorami komponenty należące do tego samego wirtualnego formularza

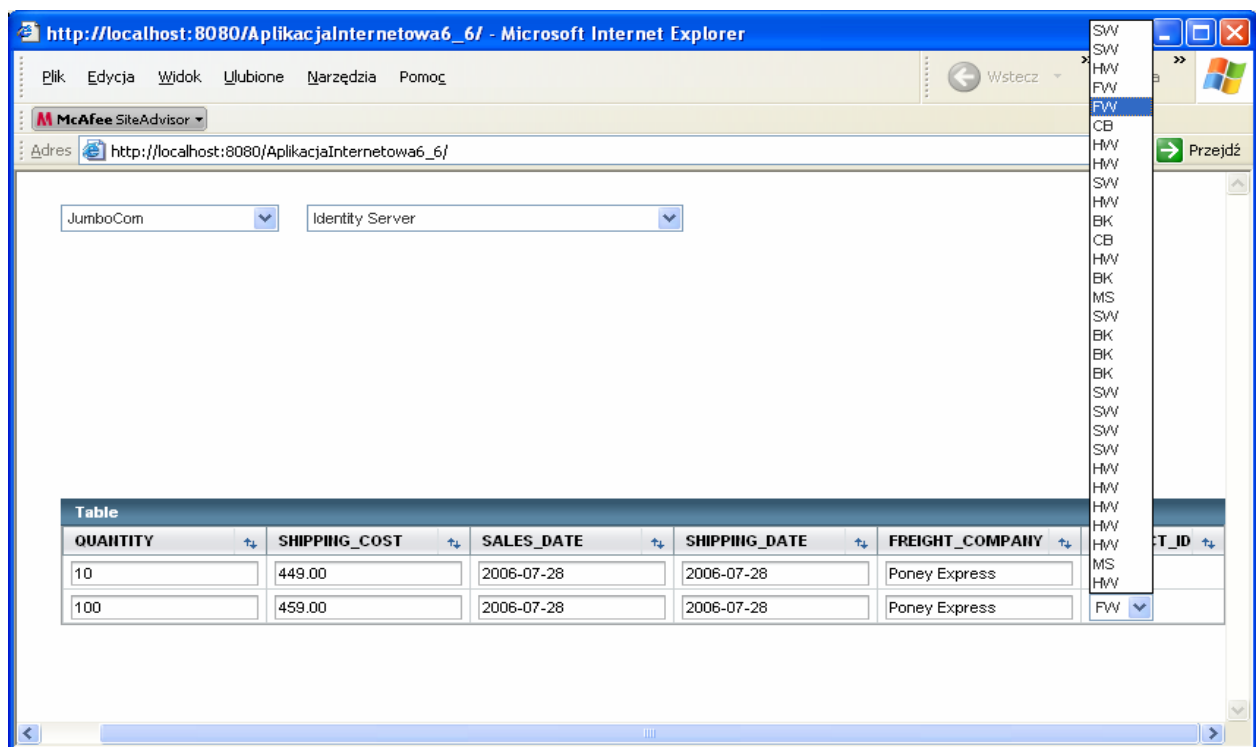




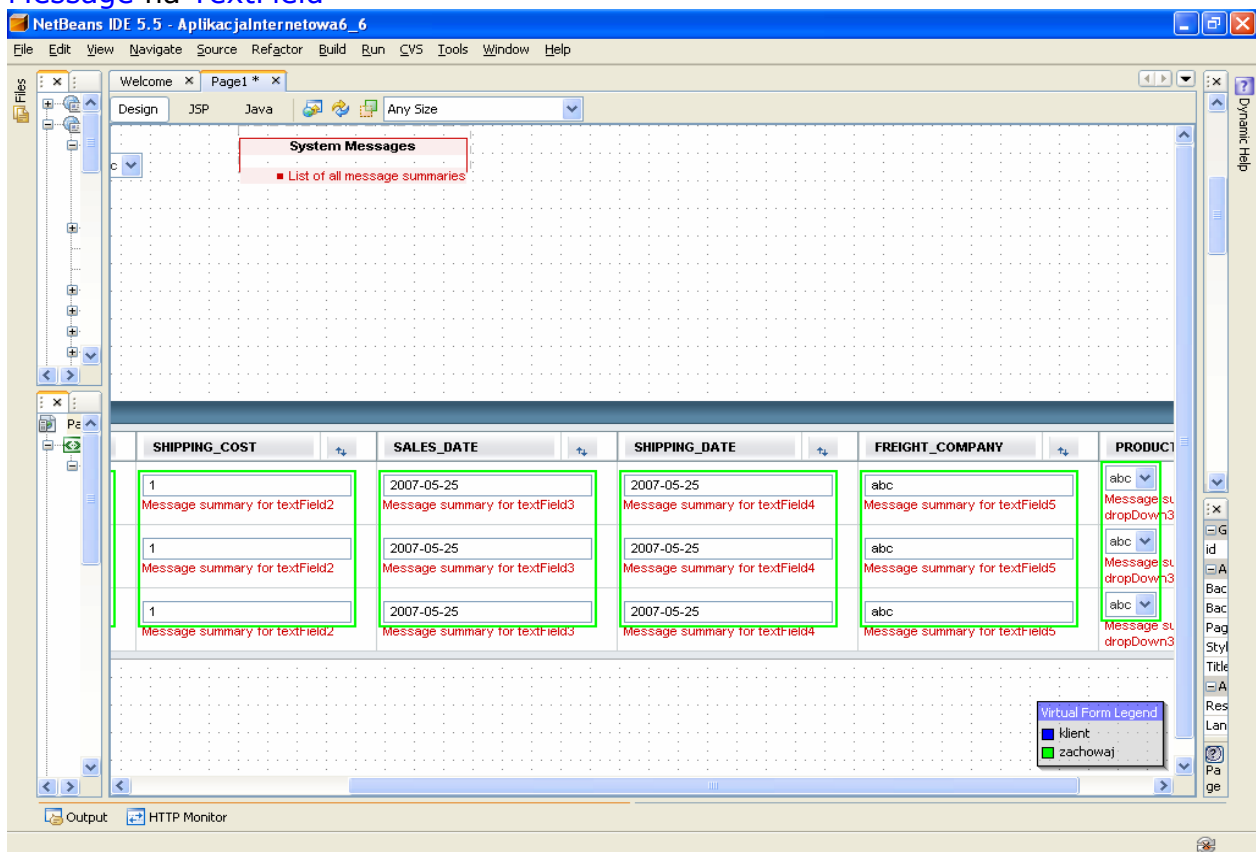
9.5. Uzupełnienie kodu źródłowego metody `dropDown1_processValueChange` dla zdarzenia typu `ProcessValueChange`
Plik Page1.java

```
public void dropDown1_processValueChange(ValueChangeEvent event)
{
    // TODO: Replace with your code
    try {
        getSessionBean1().getPurchase_orderRowSet().setObject(
            1, dropDown1.getSelected());
        purchase_orderDataProvider.refresh();
        form1.discardSubmittedValues("zachowaj");
        /*ta linia kodu powinna pojawić się po to, aby po wyborze nowego klienta ukazały się nowe
        dane w komponencie table1 */
        customerDataProvider.refresh();//odświeżenie zawartości komponentu
        //dropDown1
    } catch (Exception e) {
        error("Cannot switch to customer " +
            customerDataProvider.getValue("CUSTOMER.CUSTOMER_ID"));
        log("Cannot switch to customer " +
            customerDataProvider.getValue("CUSTOMER.CUSTOMER_ID"), e);
    }
}
```

9.6. Należy uruchomić aplikację za pomocą **Build, Deploy i Run**. Po wyborze klienta z listy **dropDown1** w wierszu komponentu **table1** ukażą się pozostałe dane tego klienta.

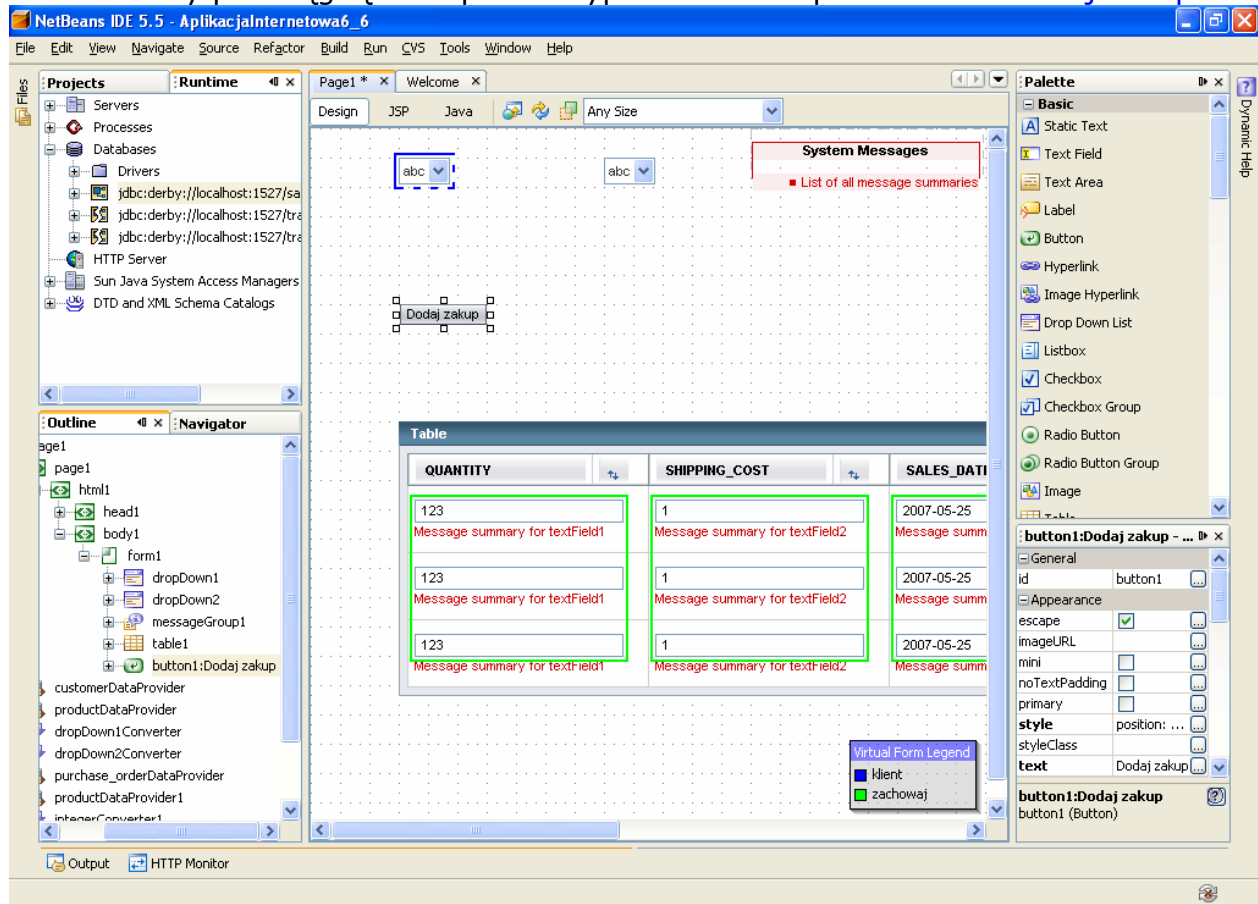


10. Należy przeciągnąć komponenty **Message** na każdy z komponentów typu **TextField** oraz **Drop Down List** i za pomocą **Ctrl+Shift** przeciągnąć komponenty **Message** na **TextField**



11. Wstawienie przycisku Dodaj zakup

- Należy przeciągnąć komponent typu **Button** i napisać na nim **Dodaj zakup**.



- Obsługa zdarzenia klawisza button1 **Dodaj zakup**. Po naciśnięciu klawisza **button1** pojawia się nowy rowek w komponencie **table1**.

Plik Page1.java

```
public String button1_action() {
```

```
    // TODO: Process the action. Return value is a navigation
```

```
    // case name where null will return to the same page.
```

```
try {
```

```
    RowKey rk = purchase_orderDataProvider.appendRow();//dodanie nowego rowka
```

```
    purchase_orderDataProvider.setCursorRow(rk);//ustawienie kursora na nowym rowku
```

```
    purchase_orderDataProvider.setValue("PURCHASE_ORDER.ORDER_NUM",  
                                       new Integer(0)); //ustawienie zerowej wartości ORDER_NUM w nowym rowku
```

```
    purchase_orderDataProvider.setValue(  
        "PURCHASE_ORDER.CUSTOMER_ID", dropDown1.getSelected());
```

```
    //ustawienie wartości CUSTOMER_ID w nowym rowku dla klienta wybranego w komponencie dropDown1
```

```
    purchase_orderDataProvider.setValue(  
        "PURCHASE_ORDER.PRODUCT_ID", dropDown3.getSelected());
```

```
    //ustawienie wartości PRODUCT_ID w nowym rowku dla produktu wybranego w komponencie dropDown3
```

```
    } catch (Exception ex) {
```

```
        log("Error Description", ex);
```

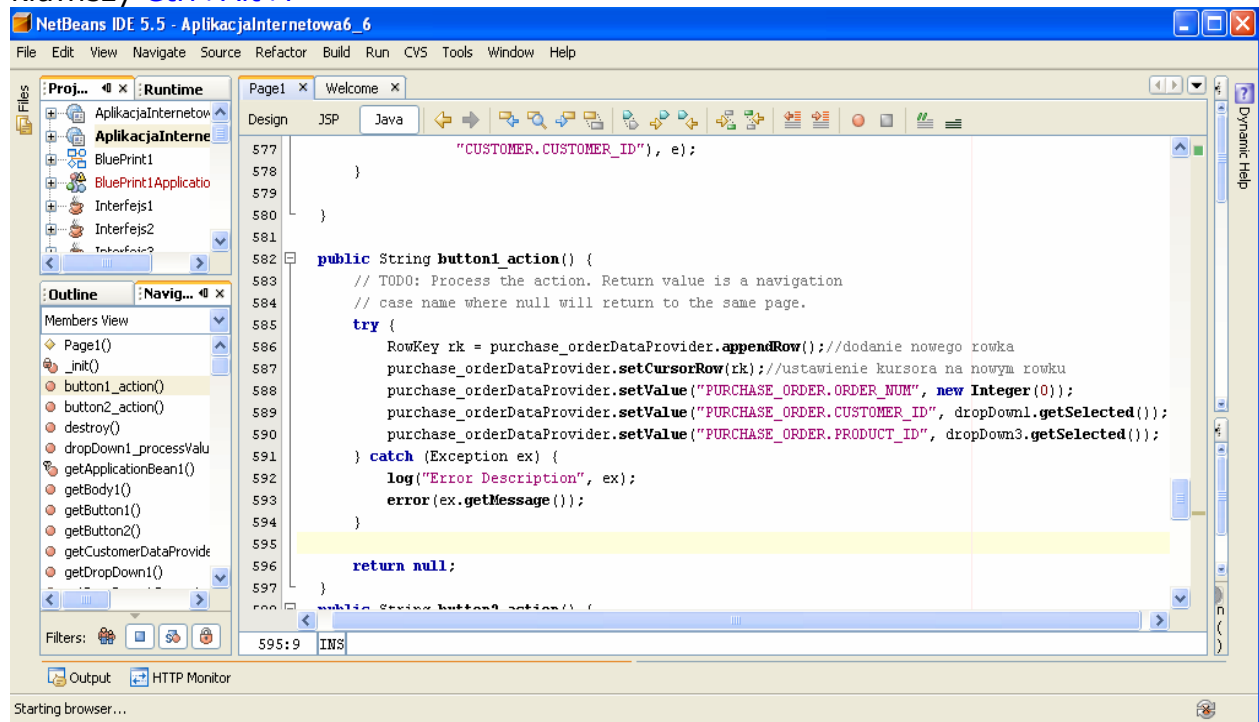
```
        error(ex.getMessage());
```

```
    }
```

```
    return null;
```

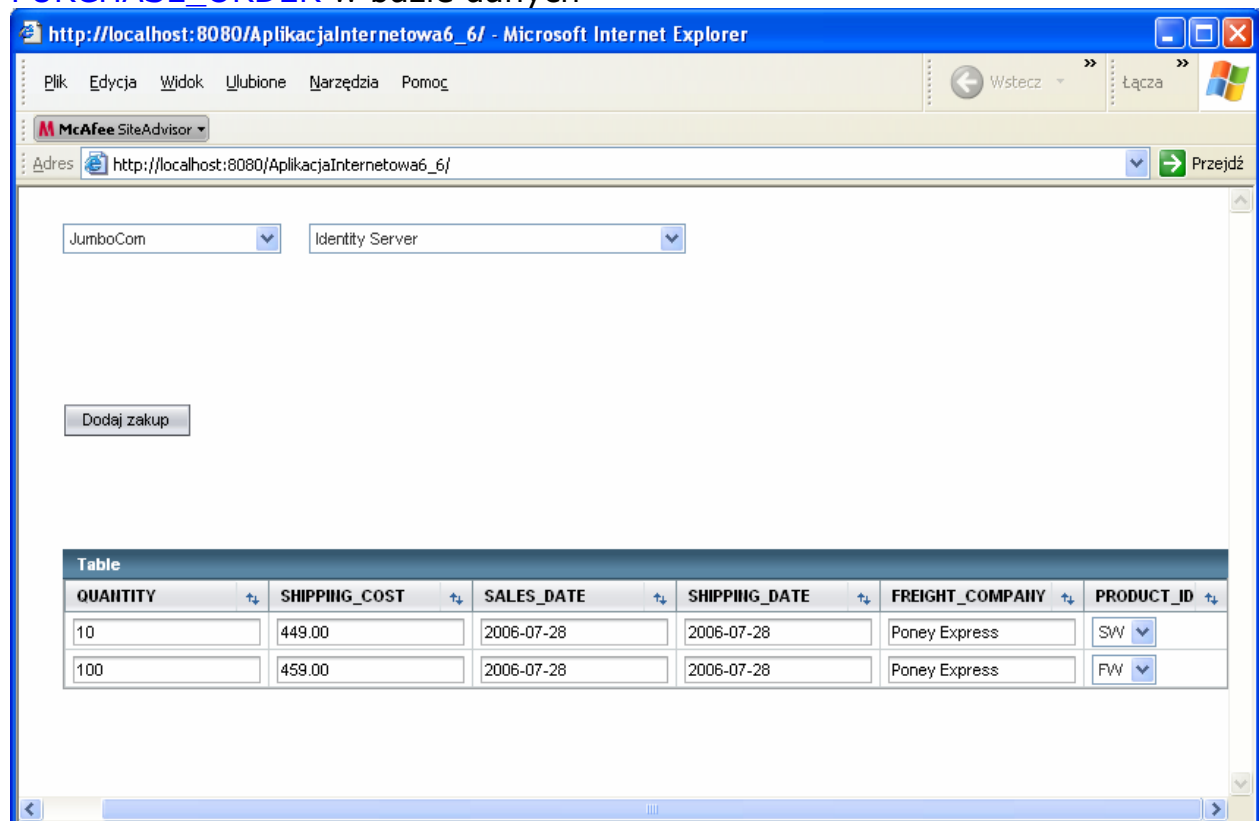
```
}
```

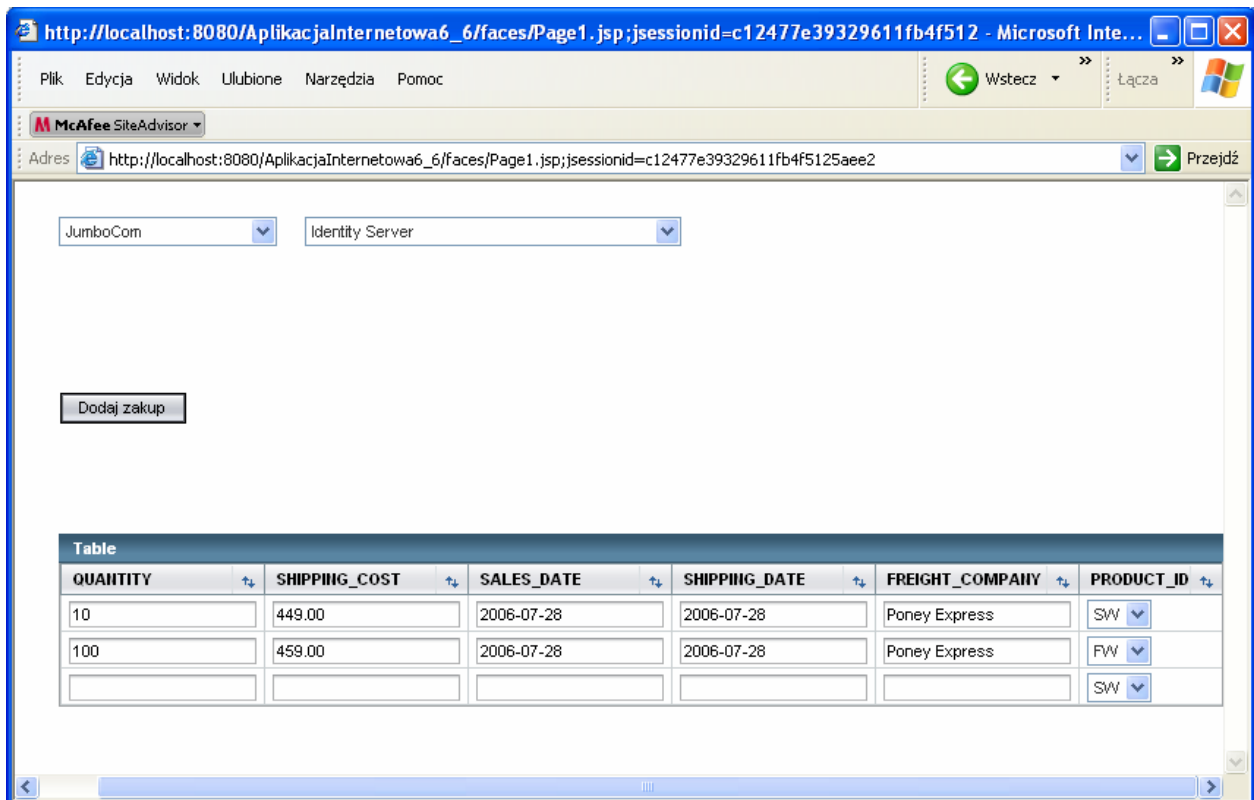
Należy wykonać import `import com.sun.data.provider.RowKey;` za pomocą klawiszy **Ctrl+Alt+F**



12. Należy uruchomić aplikację za pomocą **Build, Deploy i Run**. Po wyborze klienta z listy **dropDown1** w wierszu komponentu **table1** ukażą się jej zakupy.

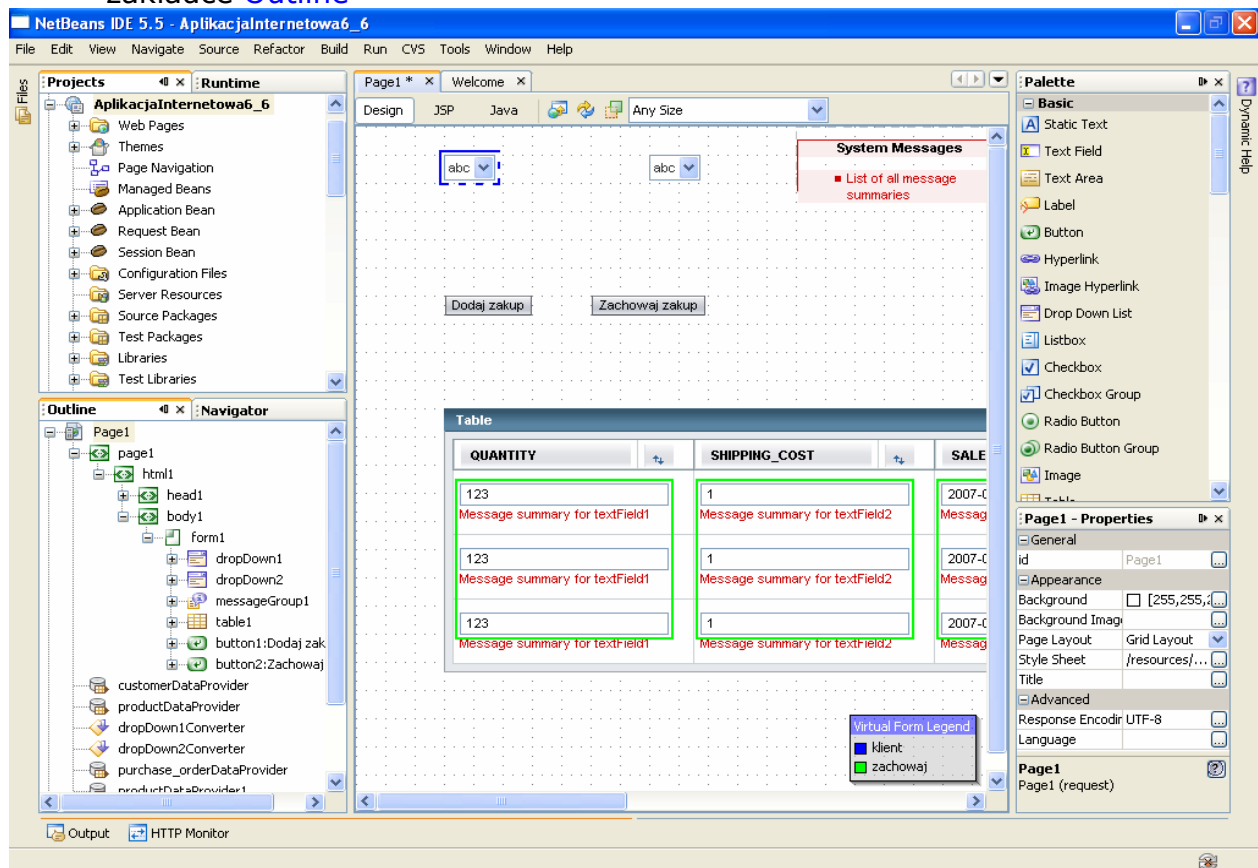
- Po kliknięciu klawisza **Dodaj zakup** ukaże się pusty rowek w komponentcie **table1**. Dane wpisane tam nie są zapisywane do tabeli **PURCHASE_ORDER** w bazie danych



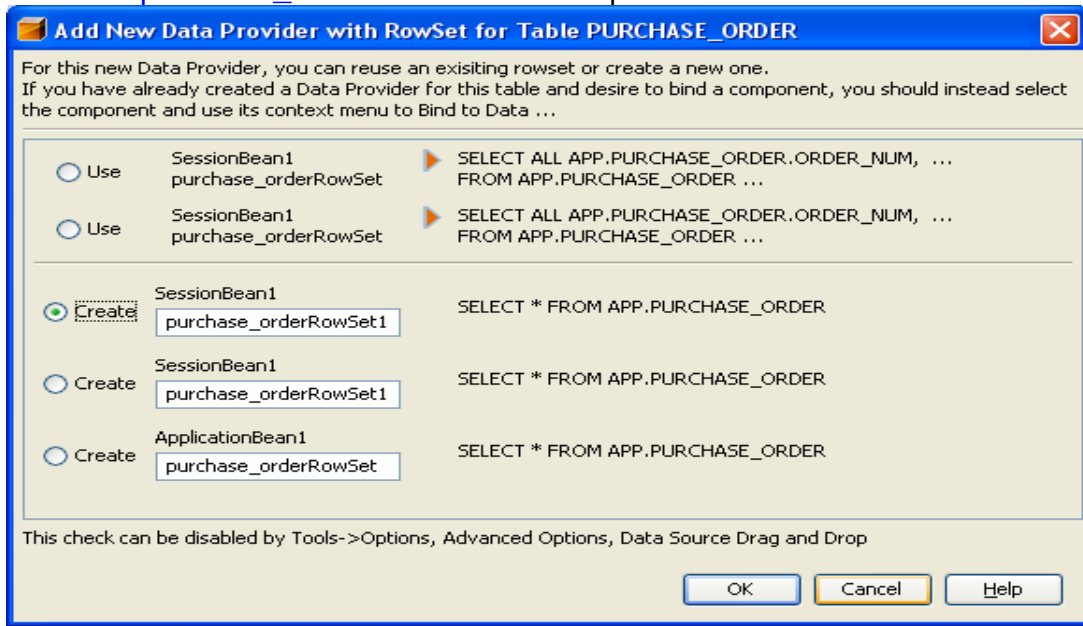


13. Zapisanie do tabeli nowego zakupu

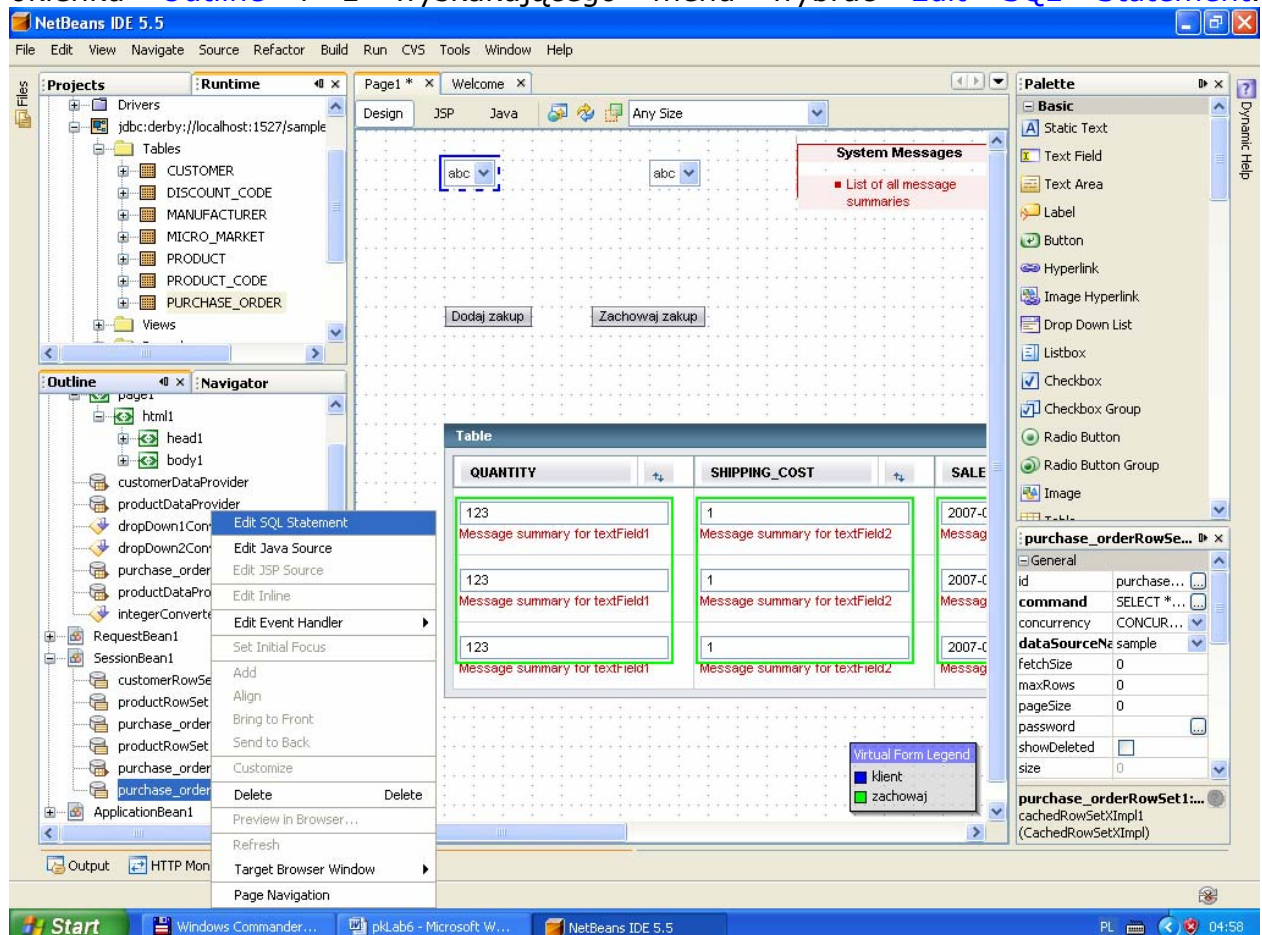
- Należy przeciągnąć klawisz **Button** i napisać na nim np. **Zachowaj zakup**
- Należy przeciągnąć tabelę **PURCHASE_ORDER** z zakładki **Runtime** i opcji **Databases > SAMPLE > Tables > PURCHASE_ORDER** na **SessionBean1** w zakładce **Outline**



- W ukazanym formularzu wybrać za pomocą klawiszy typu **RadioButton** ustawić **purchase_orderRowSet1** w komponencie **SessionBean1**

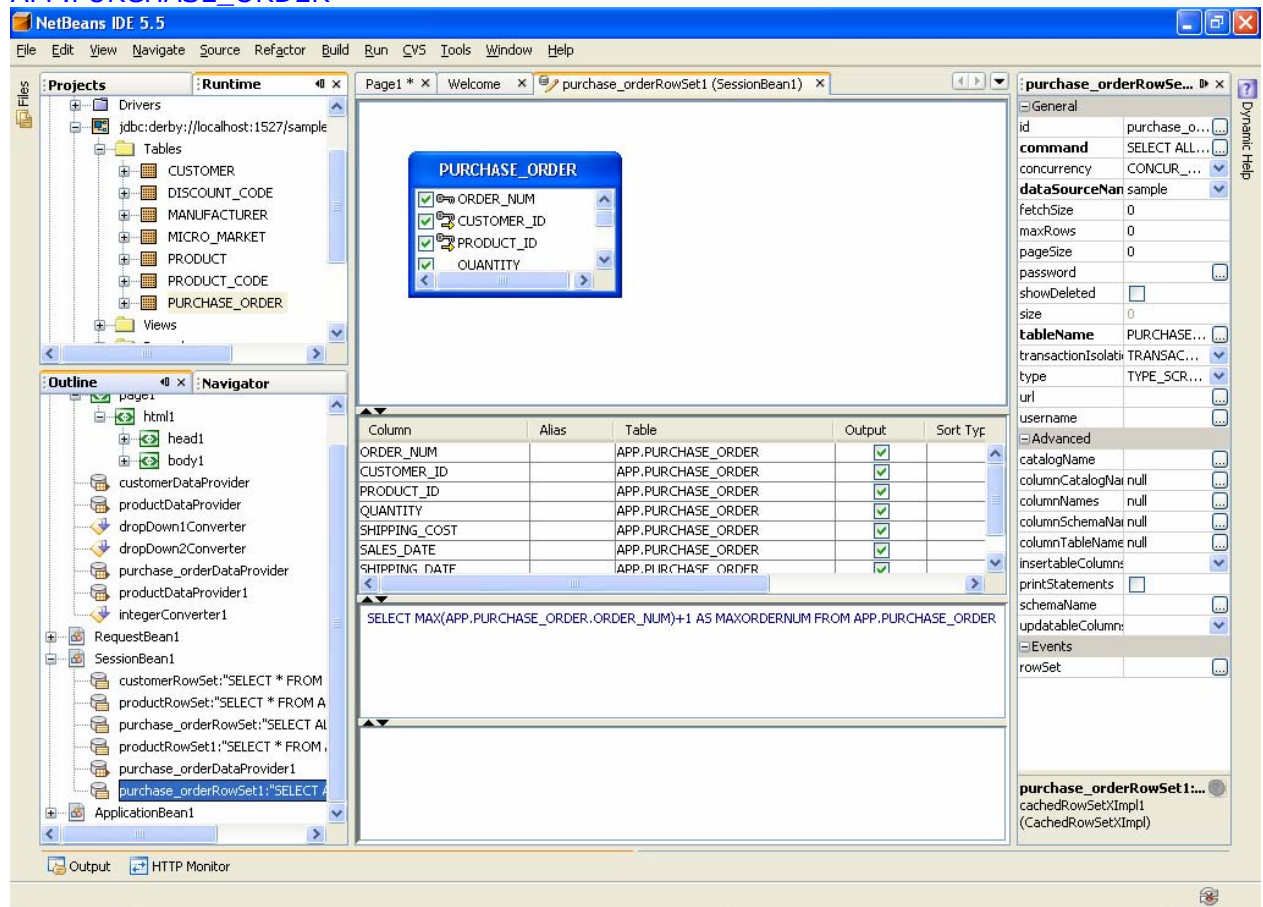


- Należy wykonać edycję zapytania **SQL** dla **purchase_orderRowSet1**. W tym celu należy kliknąć prawym klawiszem myszy na **purchase_orderRowSet1** w okienku **Outline** i z wyskakującego menu wybrać **Edit SQL Statement**.

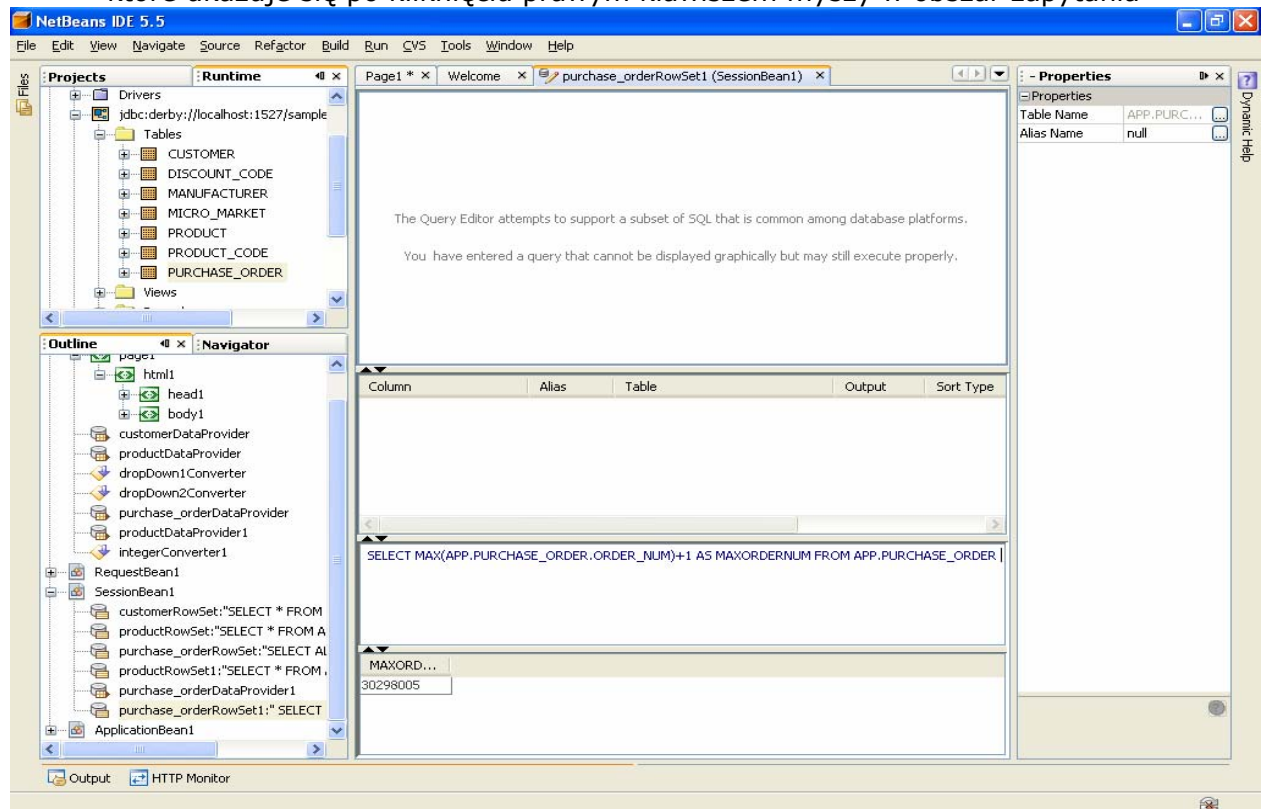


- Wpisać tekst zapytania, które wyznacza **ORDER_NUM** dla nowego zakupu (większe o 1 od największej dotąd wartości w kolumnie **ORDER_NUM** w tabeli **PURCHASE_ORDER**)

SELECT MAX(APP.PURCHASE_ORDER.ORDER_NUM)+1 AS MAXORDERNUM FROM APP.PURCHASE_ORDER



- Wykonanie zapytania za pomocą opcji **Run Query**, wybranej z wyskakującego menu, które ukazuje się po kliknięciu prawym klawiszem myszy w obszar zapytania



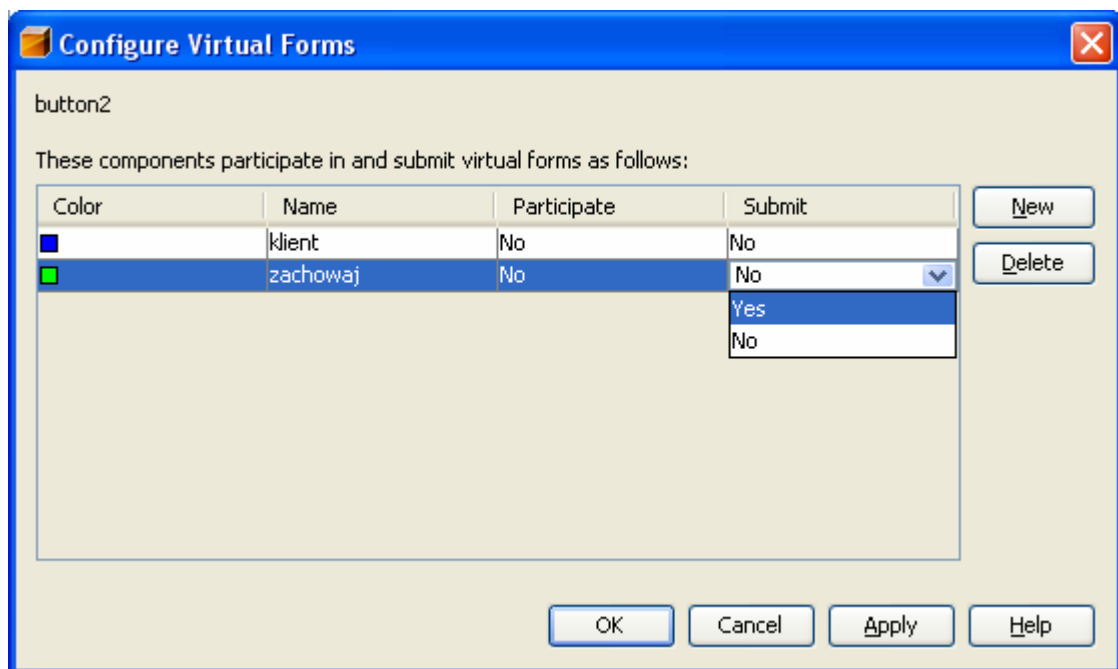
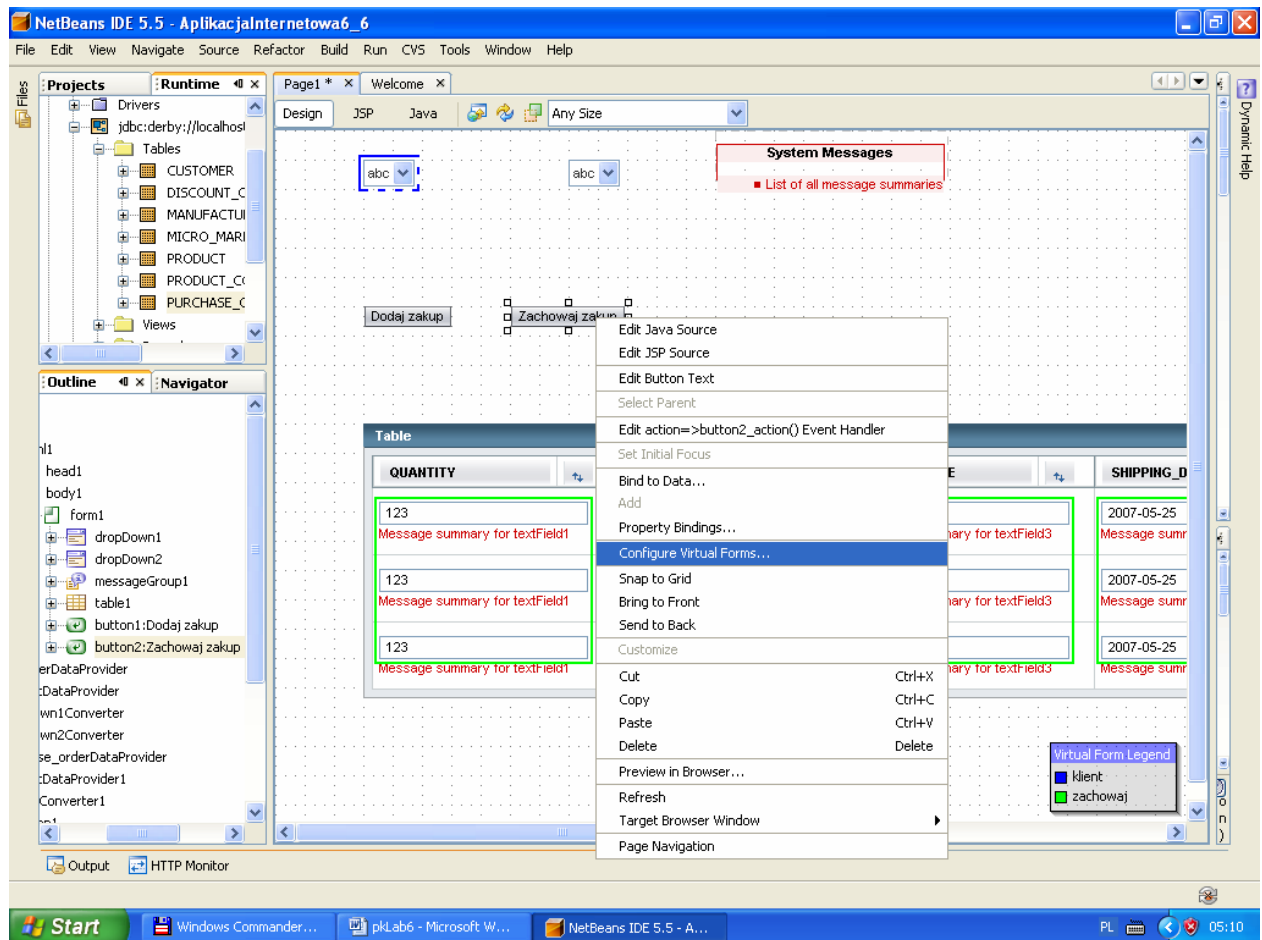
- Obsługa zdarzenia naciśnięcia klawisza `button2` (Zachowaj zakup) w pliku `Page1.java`

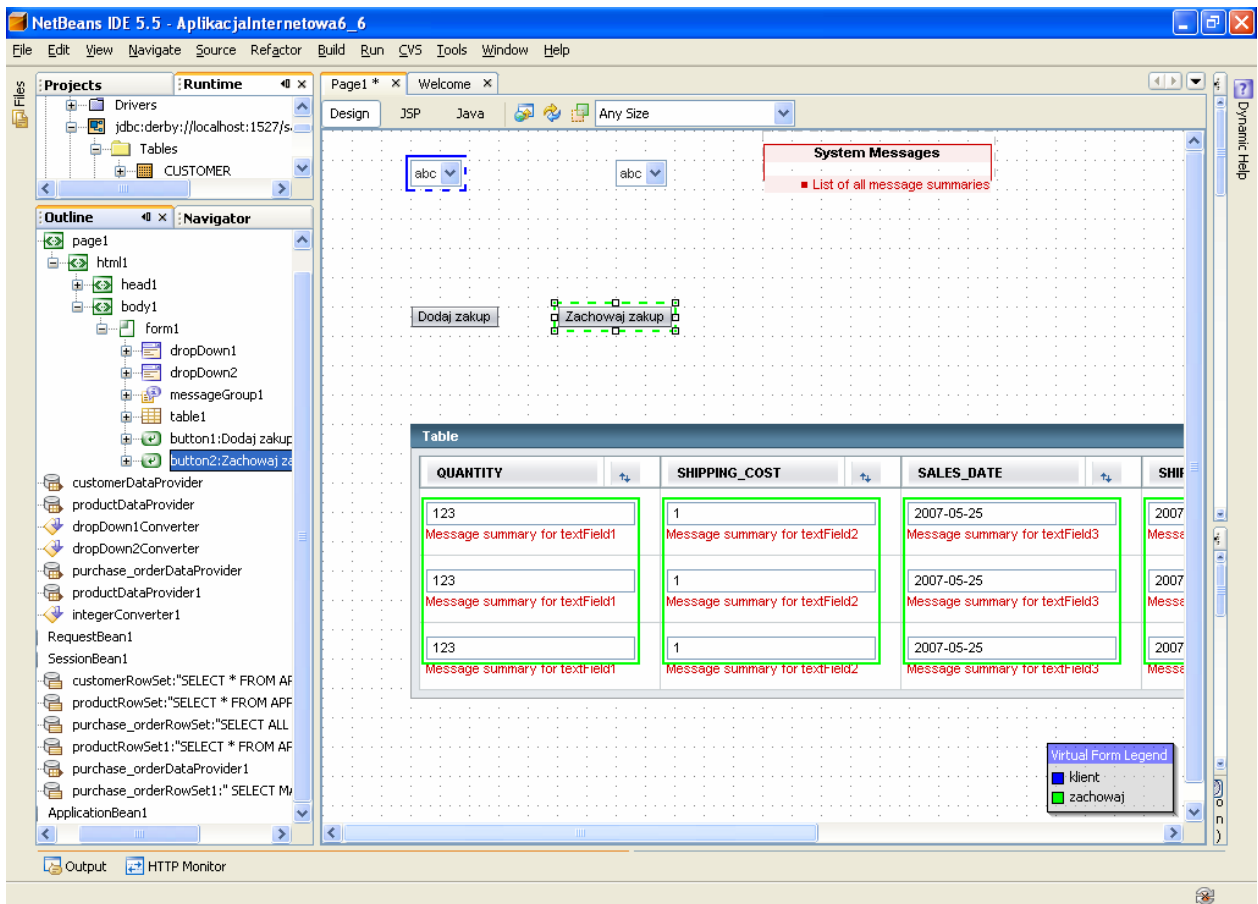
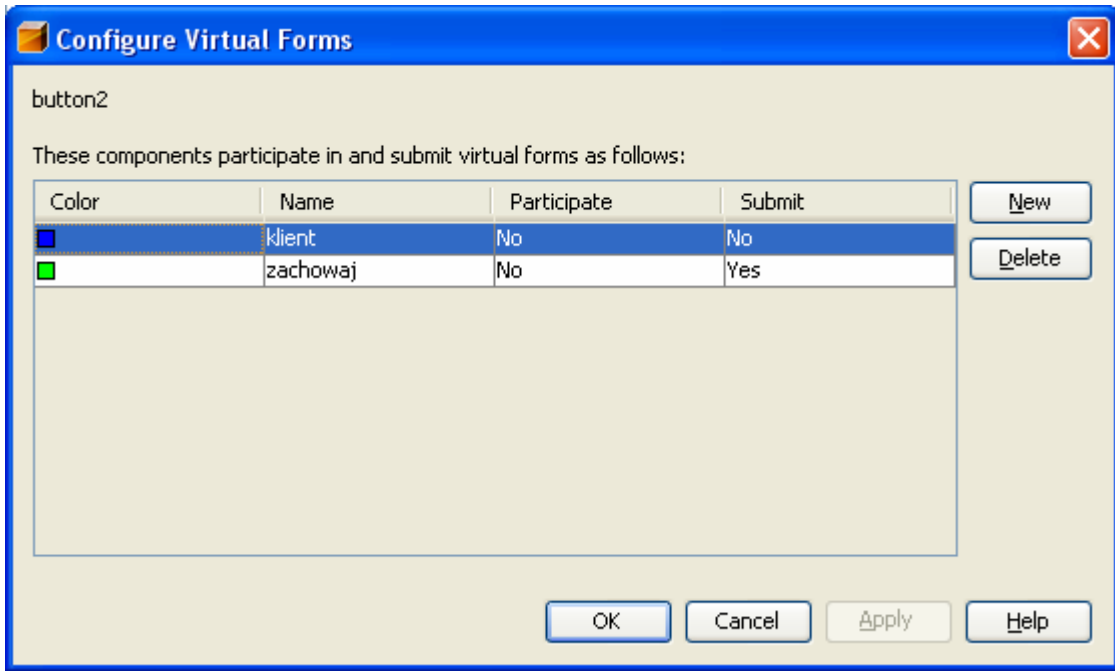
```

public String button2_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.
    try { // Get the next key, using result of query on purchase_orderDataProvider1()
        CachedRowSetDataProvider maxOrderNum =
            getSessionBean1().getPurchase_orderDataProvider1();
        maxOrderNum.refresh();
        maxOrderNum.cursorFirst();
        int nowyNumOrder //wyznaczenie poprawnej wartości kolumny ORDER_NUM dla nowego zakupu
            =((Integer)maxOrderNum.getValue("MAXORDERNUM")).intValue();
        // Navigate through rows with data provider
        if (purchase_orderDataProvider.getRowCount() > 0)
        { purchase_orderDataProvider.cursorFirst();
          do {
            if (purchase_orderDataProvider.getValue(
                "PURCHASE_ORDER.ORDER_NUM").equals
                (new Integer(0))) //po znalezieniu nowego rowka z ORDER_NUM równym 0
            { purchase_orderDataProvider.setValue(
                "PURCHASE_ORDER.ORDER_NUM",
                new Integer(nowyNumOrder));
              //wstawienie poprawnej wartości ORDER_NUM obliczonej w zapytaniu wykonanym w obiekcie
              //purchase_orderRowSet1
              ++nowyNumOrder; }
            //wyznaczenie nowej wartości ORDER_NUM dla kolejnego rowka, jeśli wprowadzono więcej niż jeden zakup
          } while (purchase_orderDataProvider.cursorNext());
        };
        purchase_orderDataProvider.commitChanges(); //wprowadzenie do bazy wszystkich
        //zmian, nie tylko wynikających z nowo wprowadzonego rowka lub kilku nowych rowków
        customerDataProvider.refresh(); //odświeżenie zawartości komponentu dropDown1
    } catch (Exception ex) {
        log("Error Description", ex);
        error("Error :"+ex.getMessage()); }
    return null; }

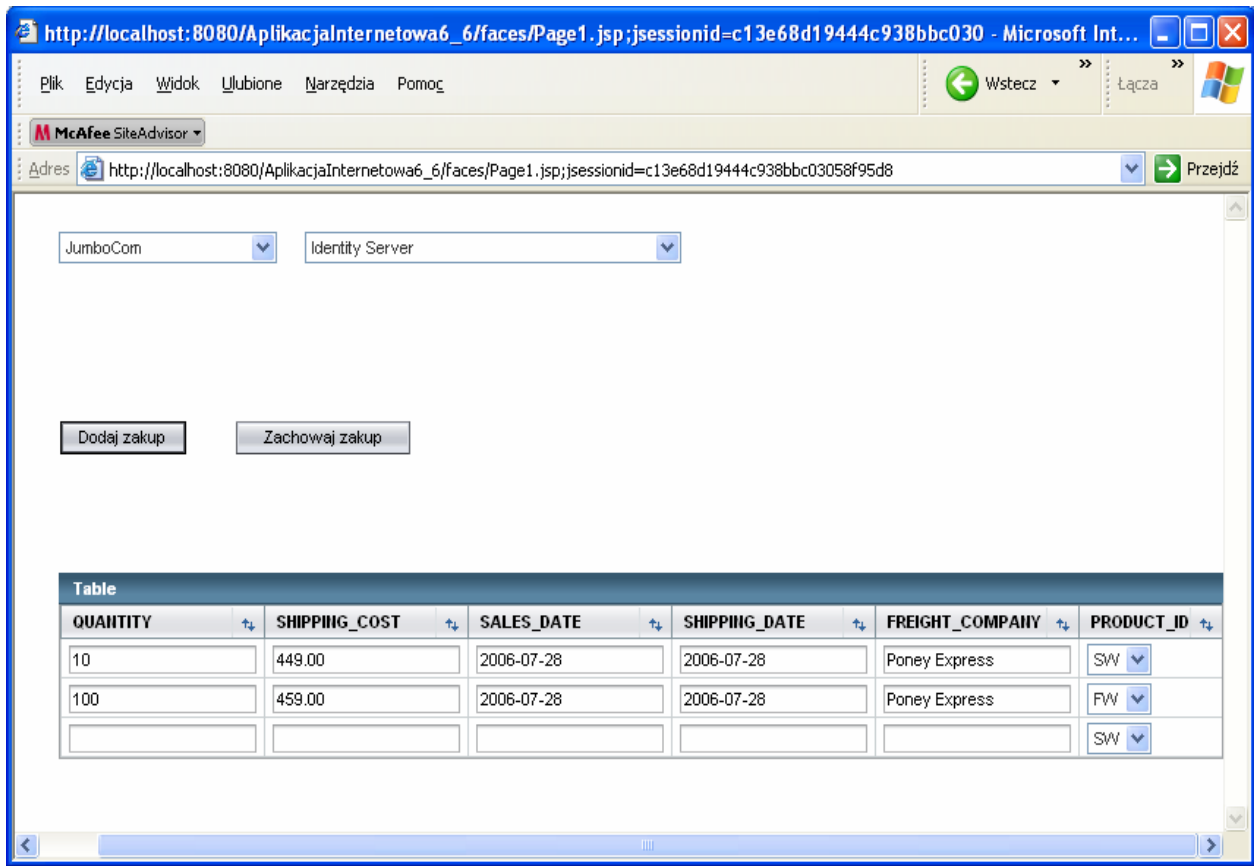
```

14. Wstawienie klawisza **button2** do wirtualnego formularza **zachowaj**





15. Uruchomienie aplikacji: Build, Deploy i Run



http://localhost:8080/AplikacjaInternetowa6_6/faces/Page1.jsp;jsessionid=c13e68d19444c938bbc030

Plik Edycja Widok Ulubione Narzędzia Pomoc

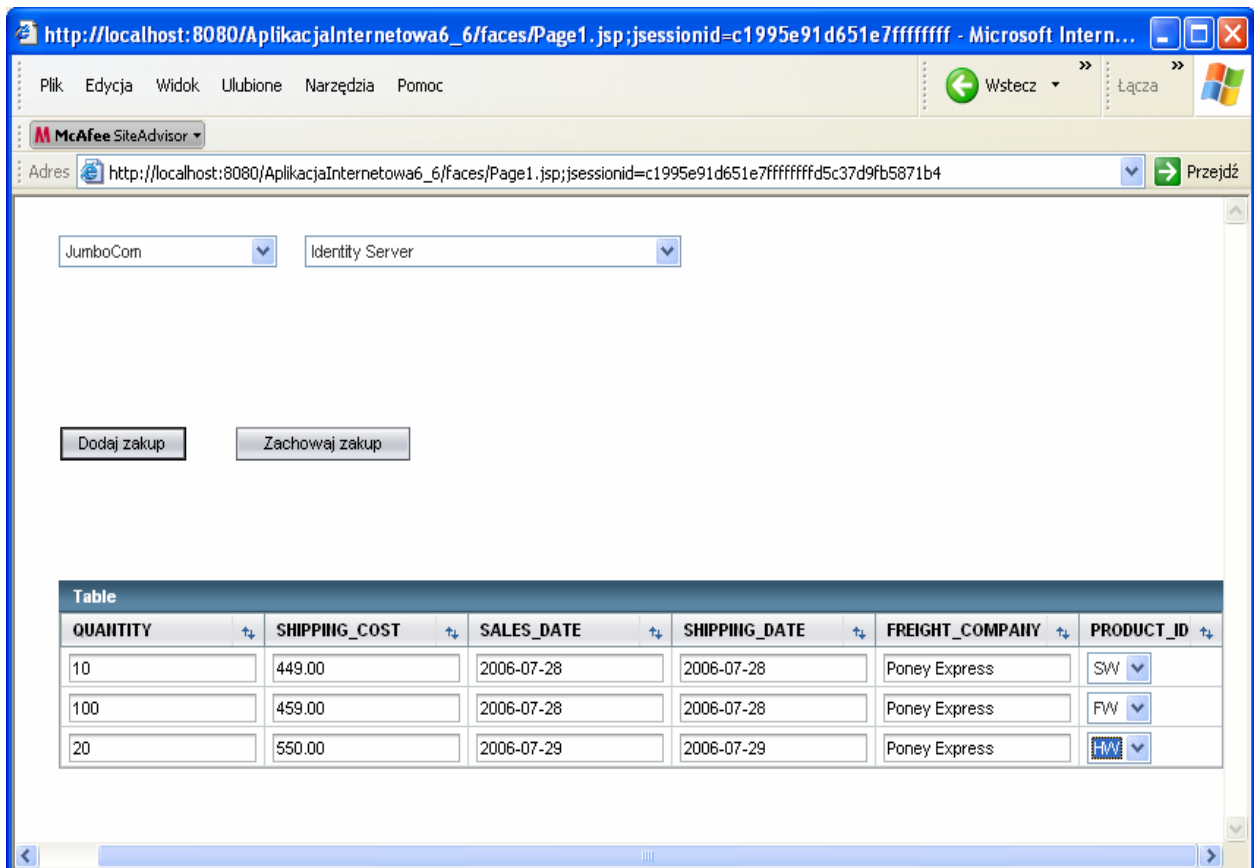
McAfee SiteAdvisor

Adres http://localhost:8080/AplikacjaInternetowa6_6/faces/Page1.jsp;jsessionid=c13e68d19444c938bbc03058f95d8

JumboCom Identity Server

Dodaj zakup Zachowaj zakup

QUANTITY	SHIPPING_COST	SALES_DATE	SHIPPING_DATE	FREIGHT_COMPANY	PRODUCT_ID
10	449.00	2006-07-28	2006-07-28	Poney Express	SW
100	459.00	2006-07-28	2006-07-28	Poney Express	FW
					SW



http://localhost:8080/AplikacjaInternetowa6_6/faces/Page1.jsp;jsessionid=c1995e91d651e7ffffff

Plik Edycja Widok Ulubione Narzędzia Pomoc

McAfee SiteAdvisor

Adres http://localhost:8080/AplikacjaInternetowa6_6/faces/Page1.jsp;jsessionid=c1995e91d651e7ffffffd5c37d9fb5871b4

JumboCom Identity Server

Dodaj zakup Zachowaj zakup

QUANTITY	SHIPPING_COST	SALES_DATE	SHIPPING_DATE	FREIGHT_COMPANY	PRODUCT_ID
10	449.00	2006-07-28	2006-07-28	Poney Express	SW
100	459.00	2006-07-28	2006-07-28	Poney Express	FW
20	550.00	2006-07-29	2006-07-29	Poney Express	FW

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