



## Vežba 4

---

*Autori: Branko Terzić, Milan Čeliković*

# Sadržaj

- Izvršavanje elemenata integracionog projekta
- Debugging
- Pregled procesa integracije
- Procedure
- Promenljive
- Sekvencer
- Korisnički definisane funkcije

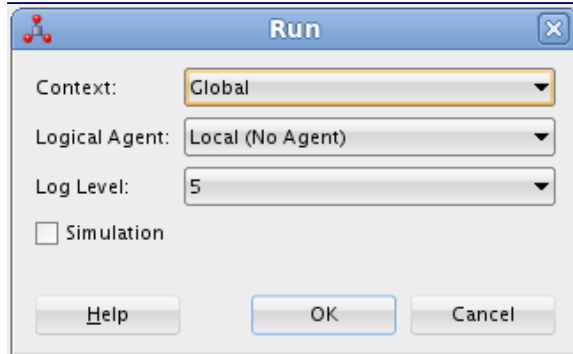
# Izvršavanje elemenata integracionog projekta

---



- Prilikom deklarativnog definisanja nekog od elemenata integracionog procesa (npr. mapiranja) ODI u pozadini generiše programski kod
- Generisani kod se izvršava u trenutku pokretanja kreiranog elementa
  - Pojedinačno, ili u okviru paketa
- Izvršavanje generisanog programskog koda rezultuje realizacijom operacija koje omogućavaju postizanje željenog rezultata
- Pokretanje izvršavanja, rezultuje kreiranjem nove sesije u kojoj se izvršava generisani kod

# Izvršavanje elemenata integracionog projekta



## – Context

- Kontekst u kome se dati element izvršava
- Podrazumevani je globalni kontekst (Development, Testing, Production)

## – Logical Agent

- Agent proces, koji treba da izvrši generisani kod

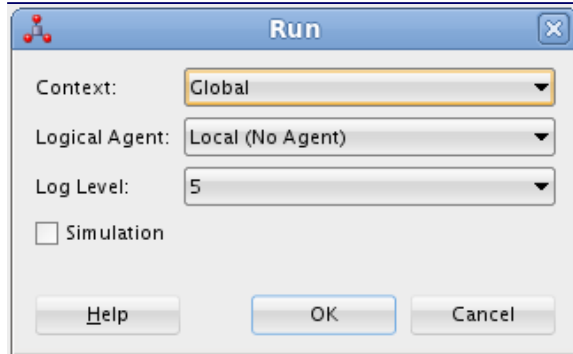
## – Log Level

- Nivo detaljnosti zapisa o koracima izvršavanja

## – Simulation

- Simulacija toka izvršavanja bez kreiranja konkretne sesije

# Izvršavanje elemenata integracionog projekta



## – Simulation

- Ne izvršava se odabrani element
- Ne kreira se nova sesija za izvršavanje elementa
- Simulira se izvršavanje elementa
- Kreira se dokument koji sadrži korake izvršavanja generisanog koda u obliku izveštaja
- Može se koristiti za pregledanje generisanog koda (Code review)

# Sadržaj

---

- Izvršavanje elemenata integracionog projekta
- Debugging
- Pregled procesa integracije
- Procedure
- Promenljive
- Sekvencer
- Korisnički definisane funkcije

# Debugging

---

- Podrazumeva kontrolisano izvršavanje generisanog programskog koda prilikom pokretanja sesije
- Poseban slučaj krišćenja alata
  - Omogućava postavljanje tačaka prekida
  - Pregleda podataka izvorne ili odredišne baze podataka, koji učestvuju u izvršavanju datog elementa
  - Pregled vrednosti promenljivih
  - Pregled aktivnih niti koje izvršavaju definisane operacije

# Debugging

## **Add Breakpoint**

- Postavljanje tačke prekida

## **Start Debug Session**

- Pokretanje debug sesija za element koji je otvoren u editoru

## **Connect to Debug Session**

- Povezivanje na već započetu sesiju za debug

## **Disconnect Debug Session**

- Prekidanjem debug sesije, proces nastavlja dalje da se izvršava normalno

## **Current Cursor**

- Trenutno izvršavana linija u debug editoru

## **Get Data**

- Prikaz SQL koda koji će se izvršiti nad izvornim odnosno odredišnim tabelama, prilikom izvršavanja naredbe koja se izvršava kao sledeća (moućnost zadavanja SQL upita)



# Debugging

---

## Step into

- Pristup podčvoru trenutno izvršavanog čvora

## Run to Task End

- Izvršavanje se nastavlja do kraja započete operacije

## Run to Next Task

- Izvršavanje se nastavlja do početka sledeće operacije

## Run to Step End

- Izvršavanje se nastavlja do kraja započetog koraka

## Run to Next Step

- Izvršavanje se nastavlja do početka sledeće operacije

## Pause

## Resume

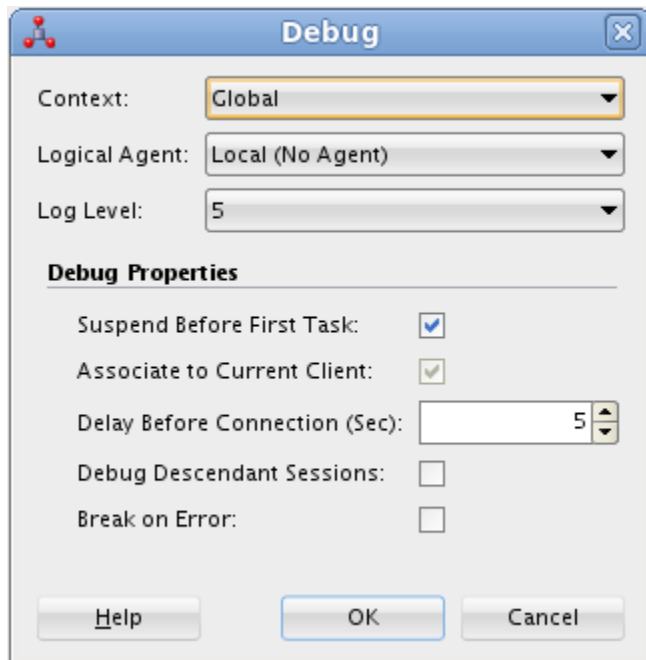
# Debugging

---

- Postavljanje tačaka prekida
  - Nije moguće kod mapiranja
  - Kod paketa
    - Na svaki od koraka izvršavanja
  - Kod procedura
    - Na svaku od pojedinačnih operacija (**Task**)
- Nivo detaljnosti
  - Nivo izvršavanja ODI atomičke operacije
  - Odnos koraka izvršavanja **Step - Task**
    - Parent – Child

# Debugging

- Pokretanje debug sesije
  1. Desnim klikom na mapiranje, proceduru ili paket odabrati opciju **Debug**
  2. Popuniti **Debug** opcije



# Debugging

---

## – Suspend Before First Task

- Zaustaviti izvršavanje pre prve operacije

## – Associate to Current Client

- Da li alat treba da otvori debug „Editor“

## – Delay Before Connection (sec)

- Broj sekundi pre nego što Debugger treba da se poveže na pokrenutu sesiju

## – Debug Descendant Sessions

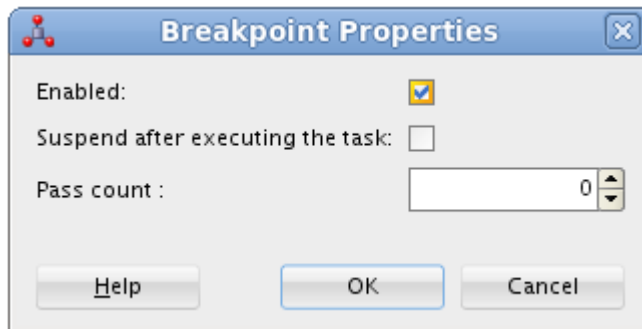
- Odnosi se na procedure i pakete
- Da li alat treba da pita korisnika da li pri pokretanju podseseije za aktivnu sesiju, treba da započne novu debug sesiju za podseseiju

## – Break On Error

- Da li debugger treba da se zaustavi kada se naiđe na grešku prilikom procesa debug-ovanja

# Debugging

- Postavljanje tačke prekida
  - Run > Breakpoint > Add Breakpoint
- Postavljanje opcija za tačku prekida
  - Run > Breakpoint > Edit Breakpoint



- **Enabled** – da li je tačka prekida aktivna
- **Suspend after executing task** – deaktiviraj tačku prekida nakon završavanja naredbe
- **Pass count** – koliko puta se tačka prekida neće aktivirati, pre nego što bude aktivirana prvi put



# Debugging

---

- Uklanjanje tačke prekida
  - Desnim klikom miša na tačku prekida odabrati **Remove Breakpoint**

# Debugging

---

## – Debugging Data tab

- U okviru Debug Data properties tab-a debug dijaloga
- **Source Task Data**
  - Sadrži SQL kod koji će se izvršiti nad izvornim tabelama u okviru trenutne naredbe
- **Target Task Data**
  - Sadrži SQL kod koji će se izvršiti nad odredišnim tabelama u okviru trenutne naredbe
- Odabirom **Run SQL Code** opcije, korisnik može izvršavati proizvoljne SQL upite nad trenutnim stanjem u izvorišnim ili odredišnim tabelama
  - Ova mogućnost se može iskoristiti za praćenje promena nad podacima nakon izvršavanja svake pojedinačne naredbe

# Debugging

---

## – Debugging Variables tab

- Praćenje stanja promenljivih
  - Tipa
  - Vrednosti
  - Mogućnost menjanja trenutne vrednosti



# Debugging

---

## – Stop Normal and Stop Immediate

- Stop Normal
  - Sesija debug-ovanja se završava nakon završetka trenutno aktivne operacije
- Stop Immediate
  - Nasilno zautavljanje sesije

# Sadržaj

---

- Izvršavanje elemenata integracionog projekta
- Debugging
- Pregled procesa integracije
- Procedure
- Promenljive
- Sekvencer
- Korisnički definisane funkcije

# Pregled procesa integracije

---

- Za pregledanje rezultata procesa integracije koristi se **Operator Navigator** panel
  - Rezultat izvršavanja je sesija (**Session**)
    - Sesija se sastoji od koraka (Steps), kosraci od operacija(Tasks)
  - **Sessions**
    - Elementi izvršavanja grupisani prema sesijama
  - **Date**
    - Grupisanje prema datumu izvrpavanja
  - **Agent**
    - Grupisanje prema agentu koji je izvršio sesiju
  - **Status**
    - Grupisanje prema statusu sesije (Done, Error)
  - **User**
    - Grupisanje prema korisniku koji je izvršio sesiju
  - **All executions**
    - Sve do sada izvršene sesije

# Pregled procesa integracije

---

- **Operator Navigator Toolbar**



## **Refresh**

- Ručno osvežavanje prikaza Operator Navigator panela



## **Auto Refresh**

- Automatsko osvežavanje prikaza Operator Navigator panela



## **Filter**

- Filtrirane sadržaja Operator Navigator panela








## **Connector Navigator**

- Upravljanje sadržajem Operator Navigator panela
  - Import
  - Export
  - Purge

# Pregled procesa integracije

---

– Mogući statusi sesije u Operation Navigator panelu

-  **Done**
-  **Error**
-  **Running**
-  **Waiting**
-  **Warning**

# Pregled procesa integracije

---

- Za detaljno pregledanje rezultata sesije koja je završena uspešno
  - Identifikovati sesiju (Session), korak (Step), odnosno operaciju (Task) koja se izvršila uspešno
  - Dvostrukim klikom otvara se Session Editor
  - U okviru **Definition** taba
    - Prikaz podataka o sesiji
      - Generalni podaci
        - » **Session** sekcija
      - Podaci o broju unetih, obrisanih i ažuriranih slogova
        - » **Records Statistics** sekcija

# Pregled procesa integracije

---

- Za detaljno pregledanje rezultata sesije koja je završena neuspešno
  - Identifikovati sesiju (Session), korak (Step), odnosno operaciju (Task) koja se izvršila neuspešno
  - Dvostrukim klikom otvara se Session Editor
  - U okviru **Definition** taba
    - Prikaz podataka o sesiji
      - Generalni podaci
        - » **Session** sekcija
      - Podaci o grešci
        - » **Error** sekcija

# Pregled procesa integracije

---

## – U okviru **Code** taba

- Podaci o izvršenom SQL kodu
  - **Code type** sekcija
    - » Tip izvršenog koda
  - **Target code** sekcija
    - » Prikaz SQL koda koji se izvršio prilikom izvršavanja operacije koja je rezultovala greškom

## – U okviru **Connection** taba

- Podaci o izvornoj i odredišnoj tabeli nad kojima se izvršava data operacija



# Sadržaj

---

- Izvršavanje elemenata integracionog projekta
- Debugging
- Pregled procesa integracije
- Procedure
- Promenljive
- Sekvencer
- Korisnički definisane funkcije

# Procedure

---

- Predstavljaju set operacija koje treba izvršiti
- Koriste se samo u situacijama u kojima se željeni rezultat ne može postići primenom mapiranja
- Iako se transformacije nad podacima mogu izvršiti i u okviru procedura, to se ne preporučuje. Za operacije nad podacima treba koristiti mapiranja
- Procedure predstavljaju ručno pisani kod, za razliku od mapiranja koje se zadaje na deklarativni način
- Koraci u kreiranju procedure
  1. Kreirati novu proceduru
  2. Specificirati opcije (Procedure's options)
  3. Kreirati zadatke procedure (Procedure's tasks)

# Procedure

---

## – Kreiranje nove procedure u okviru projekta


1. U okviru **Designer Navigator>Procedures** desnim klikom odabrati **New Procedure** opciju
2. U okviru **Definition** sekcije specificirati
  1. Ime i označiti **Multi-Connections** ukoliko korišćenje procedure zahteva više od jedne aktivne konekcije
  2. **Multi-Connections** opcija specificira nad kojim objektima će se procedura izvršavati
    - » Čekiranjem se naznačava da će se prilikom izvršavanja procedure koristiti više od jedne konekcije za pristupanje elementima nad kojima se procedura izvršava
    - » Odabirom **Source i Target Technology**, kasnije se pruža mogućnost odabira odgovarajućih logičkih šema koje opisuju izvorne i odredišne tabele koje učestvuju u izvršavanju procedure

# Procedure


---

- 3. Use Unique Temporary Object Names** – ukoliko se procedura izvršava konkurentno, omogućava kreiranje jedinstvenih imena za privremene objekte, kako nebi došlo do konflikta
- 4. Remove Temporary Objects On Error** – brisanje privremenih objekata iako je došlo do greške prilikom izvršavanja procedure

# Procedure

3. U okviru **Options** sekcije specificirati opcije za kreiranu proceduru – ponašaju se kao parametri za pojedinačne korake izvršavanja procedure
  - Dodaju se odabirom dugmeta **Add Option** 
    - » **Name** – ime opcije
    - » **Type**
      - » **Boolean** – mogu se koristiti da bi se utvrdilo da li neka komanda u okviru procedure treba da se izvrši.
      - » **Value and Text** – prosleđivanje tekstualnih ili numeričkih parametara koje se mogu koristiti u okviru procedure, pozivom **getOption()** metode (Odi Objects sekcija u okviru Expression Editor-a)
    - » **Default Value** – podrazumevana vrednost, ukoliko nije zadata konkretna
    - » **Direct Execution Value** – zadavanje različitih vrednosti za Default Value, prilikom testiranja
    - » **Redosled** proveravanja svake od opcija odgovara redosledu njihovog navođenja

# Procedure

4. U okviru **Tasks** sekcije specificirati operacije koje procedura treba da izvrši
  - Dodati novi **Task** odabirom dugmeta **Add** 
    - » **General>>Name** – naziv operacije
    - » **General>>Cleanup** – Task se izvršava čak i kada procedura vrati grešku prilikom izvršavanja (Brisane privremenih objekata)
    - » **General>>Ignore Errors** – ukoliko ne želimo da se izvršavanje procedure zaustavi na događaj greške
    - » **General>>Log Counter** – odabirom željene operacije, broji nad koliko torki je izvršena data operacija (Insert, Update, Delete, ili Errors). Rezultate pregledati u okviru **Operator Navigator** panela
    - » **General>>Log Level** – nivo detaljnosti logovanja
    - » **General>>Log Final Command** – Log-uj podatke i o poslednje izvršenoj komandi (Odi to ne radi po default)
    - » **General>>Always Execute** – izvrši task bez obzira na vrednosti pridruženih opcija

# Procedure

---

- » **Source/Target >>Transaction** – Moguće specificirati posebnu transakciju (0-9) za pojedinačne Task-ove koji će se izvršavati paralelno, ili **Autocommit** za izvršavanje svakog od izvršenih Task-ova pojedinačno
- » **Source/Target >>Commit** – commit mod komande u okviru transakcije
- » **Source/Target >>Technology** – tehnologija u okviru koje se Task izvršava, ukoliko ništa nije specificirano koristi se tehnologija specificirana u okviru definicije procedure.
- » **Source/Target>>Command** – specifikacija komande nad izvorišnim odnosno odredišnim tabelama
- » **Source/Target>>Context** – kontekst u kome se procedura izvršava (**Global**)
- » **Source/Target>>Logical Schema** – odabir logičkih šema u okviru kojih se nalaze izvorne i odredišne tabele nad čijim podacima će se procedura izvršavati
- » **Source/Target>>Transaction Isolation** – nivo transakcione izolacije za pojedinačne Task-ove (**No Change**).

# Procedure

---

- **Binding Source and Target Data**
  - Mehanizam koji omogućava izvršavanje operacija nad svakim slogom koji vrati **SQL select** naredba
    - SQL select naredba se izvršava nad izvornim tabelama a rezultat operacije nad selektovanim slogovima se primenjuje nad dw tabelama
    - U okviru jedne operacije u okviru procedure
      - U **Source Command** editoru zadati **SQL select** naredbu
      - U **Target Command** editoru specificirati kod koji će vršiti obradu selektovanih podata (**action code**)
        - » **Insert, Update** ili **Delete** naredba
          - » Rezultati dobijeni **SQL select** naredbom referenciraju se preko imena selektovanih kolona stavljanjem znaka : ispred naziva kolone, odnosno znaka # ukoliko se nad odredišnim tabelama ne sprovodi **DML** naredba



# Procedure

- **Binding Source and Target Data – primer**

Source Technology	Oracle
Source Logical Schema	ORACLE_INVENTORY
Source Command	<pre>select PRD_ID      MY_PRODUCT_ID,        PRD_NAME    PRODUCT_NAME, from   &lt;%=odiRef.getObjectName("L","PRODUCT","D")%&gt;</pre>
Target Technology	Teradata
Target Logical Schema	TERADATA_DWH
Target Command	<pre>insert into PARTS (PART_ID, PART_ORIGIN, PART_NAME) values (:MY_PRODUCT_ID, 'Oracle Inventory', :PRODUCT_NAME)</pre>

# Procedure

---

- Izvršavanje procedure
  - Pojedinačno izvršavanje
    - Desnim klikom miša na proceduru, odabrati opciju **Run**
  - Izvršavanje u okviru paketa
    - Procedura kao korak izvršavanja u okviru definicije paketa

# Sadržaj

---

- Izvršavanje elemenata integracionog projekta
- Debugging
- Pregled procesa integracije
- Procedure
- Promenljive
- Sekvencer
- Korisnički definisane funkcije

# Promenljive

---

- Objekti koji čuvaju pojedinačne vrednosti
- Vrednosti mogu biti tipa string-a broja ili datuma
- Dve vrste
  - **Global Variables**
    - Mogu se koristiti u svim aktivnim integracionim projektima
  - **Project Variables**
    - Dostupne su samo u okviru projekta u kome su kreirane

# Promenljive

---

- Kreiranje nove promenljive u okviru projekta
  1. Desnim klikom miša na **Variables** sekciju integracionog projekta odabrati **New Variable**
  2. U okviru **Definition** sekcije zadati
    - **Name** – naziv promenljive
    - **Datatype** - Tip promenljive
      - **Alphanumeric** – Tekst maksimalne dužine 255 karaktera, može uključivati **integer** ili **decimal** brožčani tip
      - **Date** – Standardni ISO date and time format: YYYY-MM-DDThh:mm:ssZ (T razdvaja vreme i datum komponente)
      - **Numeric** – **integer** tip maksimalne dužine 10
      - **Text** – slobodni tekst

# Promenljive

---

## 2. U okviru **Definition** sekcije zadati

- **Keep History**
  - **No History** – vrednost promenljive se čuva do kraja sesije
  - **Latest value**: Oracle Data Integrator u repozitorijumu čuva poslednju vrednost koju je promenljiva imala
  - **All values**: Oracle Data Integrator čuva sve vrednosti koje je promenljiva imala
- **Secure Value**
  - Selektovati ukoliko ne želimo da se niti jedna vrednost varijable ikada sačuva (ukoliko varijabla čuva password vrednost)
- **Default Value** – podrazumevana vrednost promenljive
- **Description** – kratak opis

## 3. U okviru **History** sekcije

- Pristup istoriji promena promenljive na osnovu zadatog **Keep History** parametra

# Promenljive

---

3. U okviru **Refreshing** sekcije zadati SQL upit kojim se osvežava vrednost promenljive
- **Schema** – odabrati logičku šemu nad čijim objektima će se izvršavati zadati SQL upit
  - **Select Query**
    - Zadati SQL upit koji će popuniti vrednost promenljive
      - » Koristiti **Expression Editor** za zadavanje upita 
      - » Koristiti **Test query on DBMS** za testiranje zadatog upitra 
      - » Koristiti **Refresh** za izvršavanje zadatog SQL upita i testiranje vrednosti promenljive 

# Promenljive

- **Upotreba**

1. U okviru paketa, kao korak izvršavanja u paketu

1. Prevući promenljivu u deklarisan paket

2. U okviru **Properties** taba specificirati tip promenljive **Type**

- **Declare variable**

- » Deklaracija promenljive

- **Evaluate variable**

- » Upoređivanje vrednosti promenljive sa specificiranom vrednošću (uslovno grananje u izvršavanju koraka deklarisanih u okviru paketa), if else uslov

- **Set variable**

- » Postavljanje vrednosti promenljive na zadatu vrednost

- » Inkrementiranje vrednosti promenljive za zadatu vrednost

- **Refresh variable**

- » Postavljanje vrednosti promenljive izvršavanjem specificiranog SQL upita u okviru definicije promenljive



# Promenljive

- **Upotreba**

2. U okviru elementa koji učestvuje u kreiranju mapiranja

- Filter, Join, Lookup
- U okviru **Expression Editor** prozora **ODI objects**
  - Prevuci promenljivu, promenljiva je referencirana znakom #

Attribute Expression	'#PRODUCT_PREFIX'    PR.PRODUCT_CODE	Concatenates the current project's product prefix variable with the product code. As the value of the variable is substituted, you need to enclose the variable with single quotes because it returns a string.
Join Condition	CUS.CUST_ID = #DEMO.UID * 1000 + FF.CUST_NO	Multiply the value of the UID variable of the DEMO project by 1000 and add the CUST_NO column before joining it with the CUST_ID column.
Filter Condition	ORDERS.QTY between #MIN_QTY and #MAX_QTY	Filter orders according to the MIN_QTY and MAX_QTY thresholds.

# Promenljive

- **Upotreba**

- 3. U okviru procedure

- U izrazima pojedinačnih operacija (**Task**) u okviru

```
Insert into #DWH.LOG_TABLE_NAME
```

```
Values (1, 'Loading Step  
Started', current_date)
```

- Upis podataka u log tabelu čije ime je poznato tek u trenutku izvršavanja izraza. Ime je sačuvano u LOG\_TABLE\_NAME promenljivoj.
      - U ovakvim situacijama je bolje koristiti opcije (**Options**) same procedure koje se ponašaju kao parametri

```
Insert into <%=snpRef.getOption("LogTableName")%>  
Values (1, 'Loading Step Started', current_date)
```

- » Zatim kada kreiranu proceduru koristimo u okviru paketa kao korak izvršavanja, postavimo vrednost opcije iz kreirane promenljive preko izraza **#DWH.LOG\_TABLE\_NAME**

# Promenljive

- Praćenje vrednosti promenljivih
  - U okviru **Operator navigator panela**
    - Prilikom startovanja sesije izabratei **Log level 6**
      - Dvostrukim klikom na **Session Step** ili **Session Task** date promenljive
      - U okviru **Variables and Sequence Values**

## Variable and Sequence Values

Name	Type	Value
PROBA.test	Variable	7

- U okviru **Code** sekcije, tip i kod pridružen promenljivoj

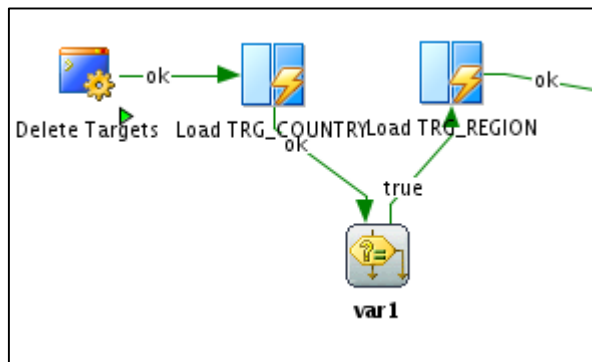
```
[-] Target Code
1 Evaluate Variable PROBA.test. EQUALS 7
```

# Promenljive

## • Primer

<b>Definition</b>	<b>Variable [Project: Demo]</b>
Refreshing	
History	Name: <input type="text" value="var1"/>
Markers	Datatype: <input type="text" value="Numeric"/>
Memo	Keep History: <input type="text" value="All Values"/>
Version	Secure Value: <input type="checkbox"/>
Privileges	Default Value: <input type="text"/>
	Description: <input type="text"/>

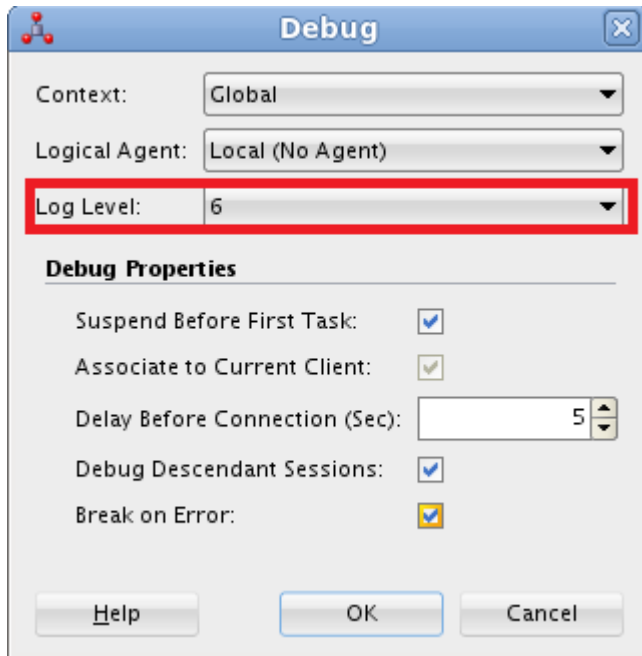
Definition	Schema: <input type="text" value="Orcl_logical_test"/>
<b>Refreshing</b>	
History	Select Query:
Markers	<code>SELECT COUNT(*) FROM &lt;%=odiRef.getObjectName("TRG_COUNTRY")%&gt;</code>
Memo	
Version	
Privileges	



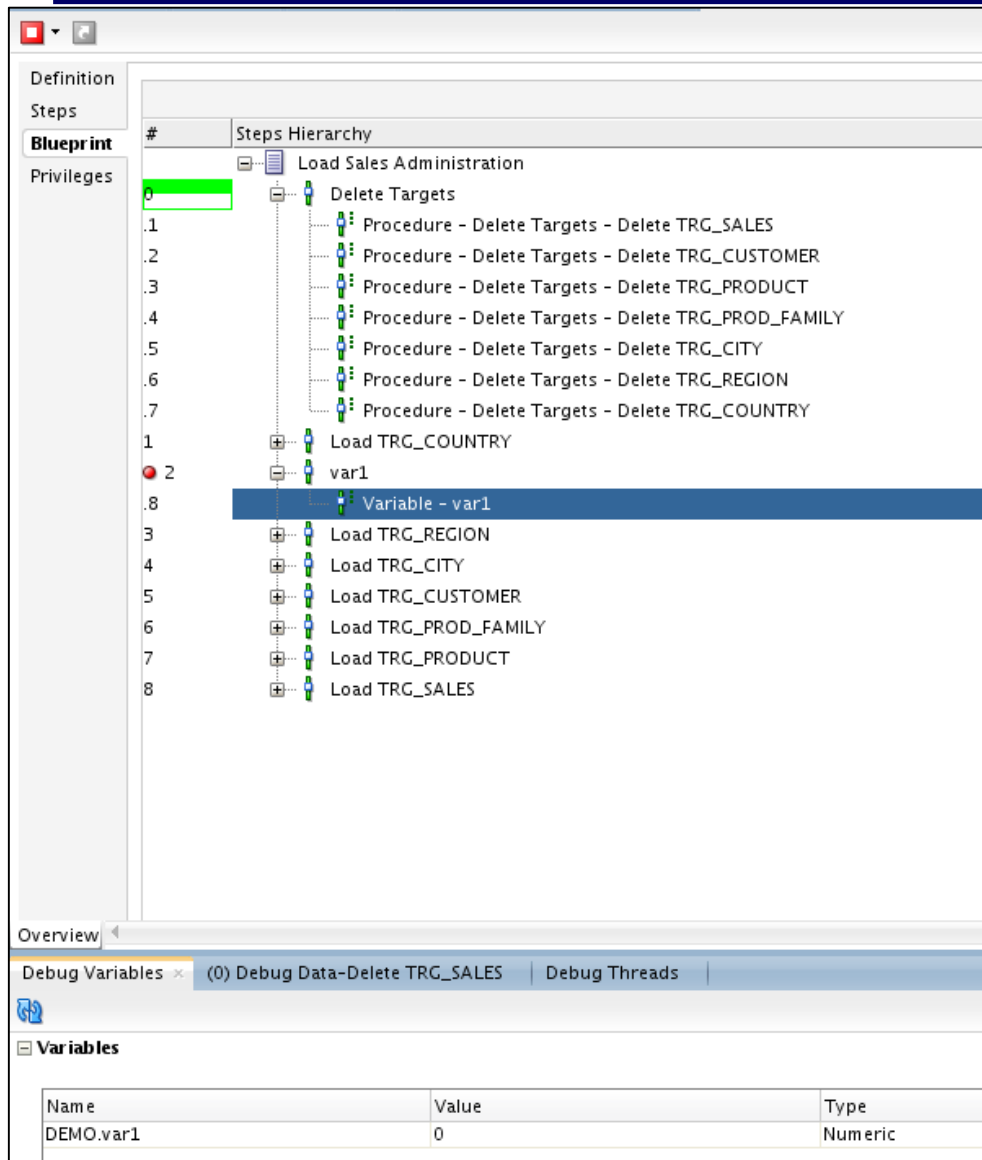
Type	<input type="text" value="Evaluate Variable"/>
Path	<input type="text" value="[Demo]"/>

var1	
Operator	Value
<input type="text" value="="/>	<input type="text" value="7"/>

# Promenljive



# Promenljive



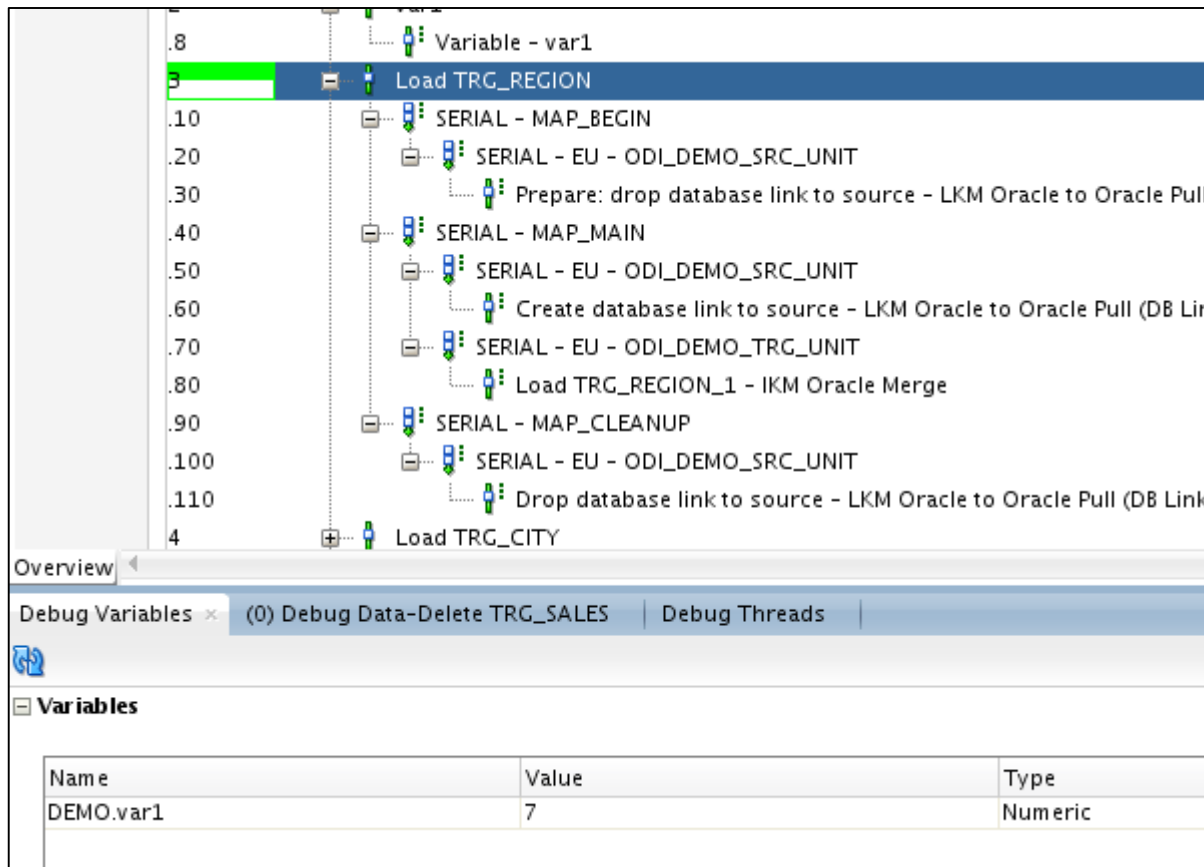
The screenshot displays a 'Steps Hierarchy' for a 'Blueprint'. The hierarchy is as follows:

- 0 Load Sales Administration
  - .1 Delete Targets
    - .1 Procedure - Delete Targets - Delete TRG\_SALES
    - .2 Procedure - Delete Targets - Delete TRG\_CUSTOMER
    - .3 Procedure - Delete Targets - Delete TRG\_PRODUCT
    - .4 Procedure - Delete Targets - Delete TRG\_PROD\_FAMILY
    - .5 Procedure - Delete Targets - Delete TRG\_CITY
    - .6 Procedure - Delete Targets - Delete TRG\_REGION
    - .7 Procedure - Delete Targets - Delete TRG\_COUNTRY
  - 1 Load TRG\_COUNTRY
  - 2 var1
    - .8 Variable - var1
  - 3 Load TRG\_REGION
  - 4 Load TRG\_CITY
  - 5 Load TRG\_CUSTOMER
  - 6 Load TRG\_PROD\_FAMILY
  - 7 Load TRG\_PRODUCT
  - 8 Load TRG\_SALES

The 'Variables' table at the bottom shows the following data:

Name	Value	Type
DEMO.var1	0	Numeric

# Promenljive



The screenshot displays the Oracle ODI interface. The top part shows a process flow diagram with several steps: Variable - var1, Load TRG\_REGION (highlighted), SERIAL - MAP\_BEGIN, SERIAL - EU - ODI\_DEMO\_SRC\_UNIT, Prepare: drop database link to source - LKM Oracle to Oracle Pull, SERIAL - MAP\_MAIN, SERIAL - EU - ODI\_DEMO\_SRC\_UNIT, Create database link to source - LKM Oracle to Oracle Pull (DB Link), SERIAL - EU - ODI\_DEMO\_TRG\_UNIT, Load TRG\_REGION\_1 - IKM Oracle Merge, SERIAL - MAP\_CLEANUP, SERIAL - EU - ODI\_DEMO\_SRC\_UNIT, Drop database link to source - LKM Oracle to Oracle Pull (DB Link), and Load TRG\_CITY.

Below the process flow, there is a 'Variables' section with a table showing the value of the variable 'var1'.

Name	Value	Type
DEMO.var1	7	Numeric

All Executions

- 20 - Load Sales Administration - Dec 8, 2015 2:49:48 AM
  - Variables
    - 0 - Delete Targets - Dec 8, 2015 2:49:48 AM
    - 1 - Load TRG\_COUNTRY - Dec 8, 2015 2:50:03 AM
    - 2 - var1 - Dec 8, 2015 2:50:22 AM
      - 8 - Variable - var1
    - 3 - Load TRG\_REGION - Dec 8, 2015 2:51:03 AM
    - 4 - Load TRG\_CITY - Dec 8, 2015 2:52:46 AM
    - 5 - Load TRG\_CUSTOMER - Dec 8, 2015 2:53:42 AM
    - 6 - Load TRG\_PROD\_FAMILY - Dec 8, 2015 2:53:43 AM
    - 7 - Load TRG\_PRODUCT - Dec 8, 2015 2:53:43 AM
    - 8 - Load TRG\_SALES - Dec 8, 2015 2:53:43 AM
  - 19 - Delete Targets - Dec 8, 2015 2:47:55 AM
  - 18 - DEMO.var1 - Dec 8, 2015 2:42:33 AM

Variable and Sequence Values		
Name	Type	Value
DEMO.var1	Variable	7



# Sadržaj

---

- Izvršavanje elemenata integracionog projekta
- Debugging
- Pregled procesa integracije
- Procedure
- Promenljive
- Sekvencer
- Korisnički definisane funkcije

# Sekvencer

---

- Promenljiva koja inkrementira samu sebe
- Može biti kreirana i dostupna na nivou projekta ili na globanom nivou
- ODI podržava tri vrste sekvencera
  - **Standard sequences**
    - Vrednosti sekvencera se čuvaju u okviru ODI repozitorijuma
  - **Specific sequences**
    - Vrednosti se čuvaju u okviru kolone tabele
  - **Native sequence**
    - Vrednosti generiše DBMS

# Sekvencer

---

- Kreiranje **Standard** sekvencera
  1. Desnim klikom na **Sequences** čvor u okviru integracionog projekta odabrati **New Sequence** opciju
  2. Uneti ime u **Name** delu
  3. Odabrati inkrement u **Increment** delu
  4. Odabrati **Standard Sequence** u **Sequence configuration**
  5. Sačuvati načinjene izmene

# Sekvencer

---

## – Kreiranje **Specific** sekvencera

1. Desnim klikom na **Sequences** čvor u okviru integracionog projekta odabrati **New Sequence** opciju
2. Uneti ime u **Name** delu
3. Odabrati inkrement u **Increment** delu
4. Odabrati **Specific Sequence** u **Sequence configuration**
  1. Odabrati **Logical** šemu na čije će se elemente primeniti kreirani sekvencer u **Schema** delu
  2. Unet naziv tabele u **Table** delu
  3. Uneti naziv kolone koja sadrži vrednost sekvencera u **Column** delu
  4. Specificirati SQL filter izraz za dobijanje vrednosti sekvencera ukoliko tabela sadrži više redova u **Filter to retrieve a single row** delu
5. Sačuvati načinjene izmene

# Sekvencer

---

## – Kreiranje **Native** sekvencera

1. Desnim klikom na **Sequences** čvor u okviru integracionog projekta odabrati **New Sequence** opciju
2. Uneti ime u **Name** delu
3. Odabrati inkrement u **Increment** delu
4. Odabrati **Native Sequence** u **Sequence configuration**
  1. Odabrati **Logical** šemu na čije elemente će se primeniti sekvencer
  2. Uneti naziv RDMBS sekvencera ili kliknuti na **Browse** dugme i odabrati ga u okviru **Native Sequence Name**
5. Sačuvati načinjene izmene

# Sekvencer

---

- **Upotreba**

- :<SEQUENCE\_NAME>\_NEXTVAL
- Prevući sekvencer iz **Project Sequencers** dela **ODI objects** sekcije u okviru **Expression editora**
- Zbog specifičnosti ostala dva tipa sekvencera, preporučuje se korišćenje **Native sekvencer-a**

# Sadržaj

---

- Izvršavanje elemenata integracionog projekta
- Debugging
- Pregled procesa integracije
- Procedure
- Promenljive
- Sekvencer
- Korisnički definisane funkcije

# Korisnički definisane funkcije

---

- Koriste se za realizaciju kompleksnih transformacija podataka prilikom realizacije procedura ili mapiranja
- Ukoliko je potrebno primeniti isti transformacioni šablon na različita mapiranja
- Kreiraju se kao globalne funkcije ili samo na nivou integracionog projekta
- Mogu pozivati druge korisnički definisane funkcije
- Ne mogu pozivati same sebe



# Korisnički definisane funkcije


---

- **Kreiranje**

1. Desnim klikom miša na **User Functions** u okviru **Designer Navigator** panela odabrati **New User Function**
2. U okviru **Definition** sekcije
  - **Name** – naziv funkcije
  - **Group** – naziv grupe, logičko grupisanje, obavezan parametar
  - **Syntax** – opis zaglavlja funkcije
3. U okviru **Implementation** sekcije
  - Zadati implementaciju funkcije

# Korisnički definisane funkcije

## • Primer

Definition	
Implementations	 <b>User Function</b>
Markers	Name: <input type="text" value="f"/>
Memo	Group: <input type="text" value="ff"/>
Version	Syntax:
Privileges	<input type="text" value="f(\$p1)"/>

Implementation	
Implementation syntax:	
<input type="text" value="UPPER(\$p1)"/>	
Linked Technologies:	
<input type="checkbox"/>	ODI Tools
<input type="checkbox"/>	OWB Runtime Repository
<input type="checkbox"/>	Operating System
<input checked="" type="checkbox"/>	Oracle
<input type="checkbox"/>	Oracle BAM

# Korisnički definisane funkcije

Attributes:

| | SQL (Oracle) Highlight: the whole expression

DISTINCT\_

**f(DISTINCT\_.COUNTRY)**

Functions:

- Aggregate
- Conversion
- Date and Time
- Keywords
- Math
- Operators

ODI objects:

- Project Functions**
  - ff
  - f
- OdiRef Functions

# Pitanja i komentari

---



# Sadržaj

---

- Izvršavanje elemenata integracionog projekta
- Debugging
- Pregled procesa integracije
- Procedure
- Promenljive
- Sekvencer
- Korisnički definisane funkcije

Kraj prezentácie

# Sistemi skladišta podataka



## Vežba 4

---