



## Vežba 3

---

*Autori: Branko Terzič, Milan Čeliković*

# Sadržaj

---

- Elementi integracionog projekta
- Kreiranje integracionog projekta
- Paketi
- Realizacija ECTL procesa

# Elementi integracionog projekta

---

- Elementi projekta
  - Folder
    - Grupiše ostale komponente u logičke celine
    - Sadrži paketa, procedure i podfoldere
  - Paket
    - Predstavlja tok izvršavanja (**Workflow**), koji se sastoji iz koraka koji čine dijagram izvršavanja (referencijaju mapiranja procedure ili promenljive)
  - Mapiranje
    - Predstavlja skup deklarativnih pravila koja opisuju način popunjavanja odredišnih skladišta podataka izvornim podacima
  - Procedura
    - Predstavlja skup pravila koja ne predstavljaju mapiranja (Purge database)

# Elementi integracionog projekta

---

- Elementi projekta
  - Promenljiva
  - Sekvenca
    - Promenljiva čija se vrenost automatski inkrementira, svaki put kada se koristi
  - Korisnički definisane funkcije
  - Knowledge Modul
  - Scenario
    - Jedinica izvršavanja izvršivih elemenata projekta

# Elementi integracionog projekta

---

- Knowledge Moduls (Projece and Global)
  - RKM (Reverse Knowledge Module)
    - Relalizacija R-E, specifičnog za svaku tehnologiju
  - LKM (Loading Knowledge Module)
    - Realizacija izdvajanja podataka, prilikom mapiranja
  - JKM (Jurnalizing Knowledge Module)
    - Realizacija pracenja promena nad izvornim podacima
  - IKM (Integration Knowledge Module)
    - Realizacija punjenja DW baze podataka
  - CKM (Check Knowledge Module)
    - Provera ograničenja, definisanih nad modelom
  - SKM (Service Knowledge Module)
    - Pristupanje elementima modela preko Web servisa

# Sadržaj

---

- Elementi integracionog projekta
- Kreiranje integracionog projekta
- Paketi
- Mapiranja
- Realizacija ECTL procesa

# Kreiranje integracionog projekta

1. U **Designer Navigator** panelu, u okviru **Projects** sekcije, odabrati **New Project**



The screenshot shows the 'Definition' tab of the 'Project' entity in the Designer Navigator. On the left, a vertical menu lists 'Markers', 'Memo', 'Version', 'Privileges', and 'Flexfields'. The main area is titled 'Project' and contains two input fields: 'Name:' and 'Code:'.

2. Zadati naziv projekta
3. Sačuvati načinjene izmene

# Sadržaj

---

- Kreiranja integracionog projekta
- Kreiranje integracionog projekta
- Paketi
- Mapiranja
- Realizacija ECTL procesa



# Paketi

---

- Objedinjuje više elemenata koje treba izvršiti u okviru alata ODI
- Sastoji se od sekvence koraka (Steps) predstavljenih u okviru dijagrama izvršavanja
  - Svaki korak može da se izvrši uspešno ili neuspešno
  - Zavosno od ishoda izvršavanja, tok izvršavanja koraka u paketu može da se grana
- Koraci izvršavanja mogu se realizovati izvršavanjem
  - Procedure
  - Mapiranja
  - Postavljanja vrednosti promenljive

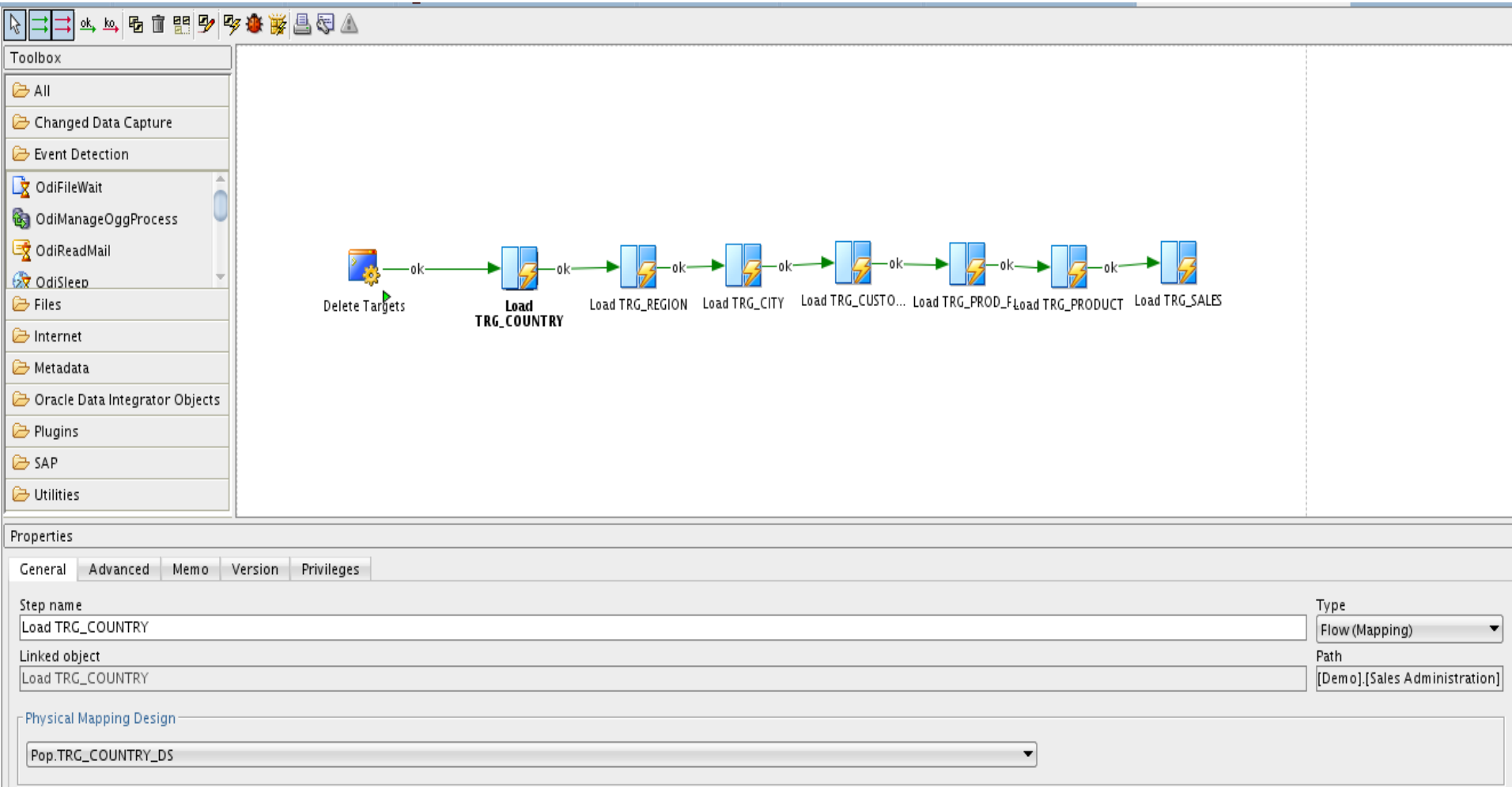
# Paketi

---

- **Elementi Package Editor alata**
  - **Package Diagram**
    - Radni prostor u kome se nalaze komponente koje predstavljaju deo kreiranog paketa (mapiranja, promenljive, modeli, tabele). Zauzima centralno mesto u okviru Package Diagram-a
  - **Package Toolbox**
    - Predstavlja listu ODI alata dostupnih za korišćenje u okviru kreiranog paketa
  - **Package Toolbar**
    - Predstavlja listu alata koje se koristite za grupisanje elemenata koji su dodati u okviru Package Diagram-a
  - **Properties Panel**
    - Sadrži opis karakteristika selektovanog elementa koji je dodat u okviru Package Diagram-a

# Paketi

- Elementi Package Editor alata



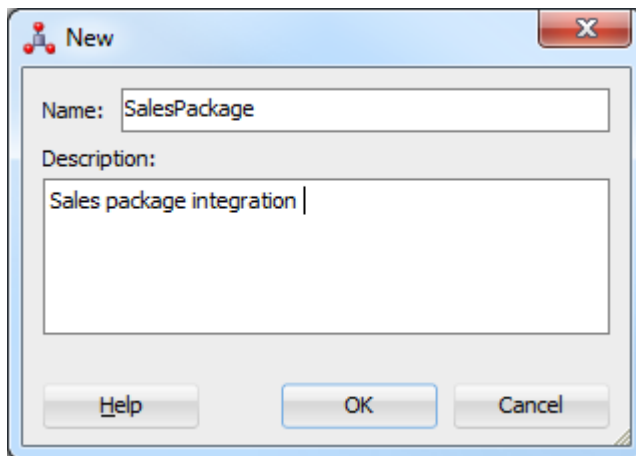
The screenshot displays the Oracle Data Integrator Package Editor interface. On the left is a **Toolbox** with categories like All, Changed Data Capture, Event Detection, and various ODI tasks such as OdiFileWait, OdiManageOggProcess, OdiReadMail, and OdiSleep. The main workspace shows a workflow diagram with the following steps: **Delete Targets**, **Load TRG\_COUNTRY**, **Load TRG\_REGION**, **Load TRG\_CITY**, **Load TRG\_CUSTO...**, **Load TRG\_PROD\_F**, **Load TRG\_PRODUCT**, and **Load TRG\_SALES**. Each step is connected to the next by a green arrow labeled "ok".

Below the workspace is the **Properties** panel, which includes tabs for General, Advanced, Memo, Version, and Privileges. The **General** tab is active, showing the following fields:

- Step name:** Load TRG\_COUNTRY
- Linked object:** Load TRG\_COUNTRY
- Physical Mapping Design:** Pop.TRG\_COUNTRY\_DS
- Type:** Flow (Mapping)
- Path:** [Demo].[Sales Administration]

# Paketi

- Kreiranje paketa
  1. U **Designer Navigator** panelu, u okviru **Projects** sekcije, u okviru podrazumevanog direktorijuma, desnim klikom na **Packages** odabrati **New Package** opciju



2. Uneti naziv i opis paketa
3. Sačuvati načinjene izmene

# Paketi

---

- Definisiranje koraka izvršavanja
  - Paket predstavlja sekvencu koraka izvršavanja koji su međusobno povezani
  - Definisiranje paketa podrazumeva definisanje koraka izvršavanja u okviru paketa
  - Koraci izvršavanja predstavljaju izvršavanje
    - Mapiranje
    - Procedura
    - Deklarisanje i definisanje promenljivih
  - Dodavanje svakog od koraka izvršavanja izvodi se krišćenjem **Drag and Drop** tehnike, prevlačenjem mapiranja, procedure ili promenljive u radni prostor paketa, pri čemu oni tada postaju jedan korak izvršavanja.

# Paketi

---

- Korak izvršavanja – Promenljive (Variable)
  - Postoji više različitih tipova koraka izvršavanja koji se odnosi na promenljive
    1. **Declare Variable** – ukoliko se promeljiva eksplicitno koristi pri izvršavanju ostalih koraka izvršavanja u okviru paketa
    2. **Refresh Variable** – ovaj tip koraka izvršavanja osvežava promenljivu izvršavanjem upita koji je specificiran u okviru deklaracije promeljive
    3. **Set Variable**
      1. **Assign** – dodela konkretne vrednosti promeljivoj
      2. **Increment** – uvećanje ili umanjenje vrenosti promenljive za zadatu vrednost
    4. **Evaluate Variable** – provera zadovoljenja logičkog uslova zadatog nad definisanom verdnošću promenljive. Koristi se prilikom grananja u izvršavanju koraka u okviru paketa

# Paketi

---

- Korak izvršavanja – brisanje
  1. Odabrati korak izvršavanja koji se želi obrisati
  2. Desnim klikom miša odabrati **Delete Step** opciju
  3. Potvrditi brisanje
  4. Sačuvati načinjene izmene
- Korak izvršavanja – dupliranje
  1. Odabrati korak izvršavanja za koji se želi napraviti kopija
  2. Desnim klikom miša odabrati opciju **Duplicate Step**
  3. Sačuvati načinjene izmene

# Paketi

---

- Korak izvršavanja – pokretanje
  1. Odabrati korak koji se želi izvršiti u okviru paketa
  2. Desnim klikom miša odabrati operaciju **Execute Step**
  3. Potvrditi na taster **OK**
  4. Rezultati izvršavanja su vidljivi u okviru **Operator Navigator** navigator panela



# Paketi

---

- Definisiranje sekvence koraka izvršavanja
  - Kada se definišu svi koraci izvršavanja, potrebno iz je povezati u lanac izvršavanja (**processing chain**)
    - Počinje tačno jednim korakom (**First Step**)
    - Svaki korak u lancu izvršavanja rezultuje jednim od dva moguća stanja: **Success or Failure**
    - Koraku koji se završio uspešno ili neuspešno može slediti sledeći korak, ili kraj izvršavanja paketa
    - Ukoliko se korak završi neuspešno može se definisati broj ponovnih pokušaja
    - Sekvenca izvršavanja paketa zapocinje tačno jednim početnim korakom izvršavanja, a može se završiti sa više završnih koraka

# Paketi

---

- Definisavanje sekvence koraka izvršavanja
  1. Desnim klikom na početni korak izvršavanja odabrati **First Step opciju**
  2. Svaki sledeći korak povezati sa predhodnim na jedan od sledćih načina
    1. Odabirom **Next Step on Success** opcije iz **Package Toolbar** sekcije
    2. Odabirom **Next Step on Failure** opcije iz **Package Toolbar** sekcije
  3. Poslednji korak od koga ne ide veza prema ni jednom drugom koraku je završni korak u lancu koraka

# Paketi

---

- Definisiranje sekvence koraka izvršavanja
  - Poslednji korak od koga ne ide veza prema ni jednom drugom koraku je završni korak u lancu koraka
    - Odabirom poslednjeg koraka u sekvenci koraka u okviru panela **Properties**
      - Odabrati **End** u okviru **Processing after failure** i **Processing after success** sekcije
        - » Sve veze koje idu od selektovanog koraka se brišu

# Paketi

---

- Pokretanje paketa
  - Desnim klikom miša na željeni paket odabrati **Run** opciju
  - Selektovati željeni paket i odabrati **Run** opciju iz ODI Toolbar sekcije
  - Selektovati željeni paket pa iz menija ODI Menu sekcije odabrati **Run>Run**
  - Rezultat izvršenja koraka nakon izvršenja paketa pogledati u okviru **Operator Navigator** panela

# Sadržaj

---

- Kreiranja integracionog projekta
- Kreiranje integracionog projekta
- Paketi
- Mapiranja
- Realizacija ECTL procesa

# Mapiranja

---

- Mapiranje predstavlja opis transformacije izvornih podataka u podatke koje treba smestiti u odredišno skladište podataka
- Mapiranje vrši čitanje i filtriranje izvornih podataka i punjenje odredišnog skladišta podataka
- Mapiranje se definiše u okviru **Mapping Editor** alata u okviru ODI okruženja
- Osnovne komponente mapiranja predstavljaju izvorne i odredišne tabele, kao i skup pravila koja opisuju način na koji se atributi izvornih tabela preslikavaju na odredišne attribute

# Mapiranja

---

- **Sekcije Mapping Editor alata**
  - **Mapping Diagram** – radni prostor u okviru koga se vrši definicija mapiranja
  - **Mapping Editor Tabs**
    - **Overview** – generalni pregled komponenti kreiranog mapiranja
    - **Logical** – predstavlja opis logičke organizacije mapiranja
    - **Physical** – predstavlja opis fizičke organizacije mapiranja
  - **Property Inspector** – prikazuje karakteristike objekta koji učestvuje u kreiranom mapiranju
  - **Component Palette** – prikazuje komponente mapiranja koje se koriste prilikom deklaracije mapiranja
  - **Structure** – Window ->Structure prikazuje strukturu mapiranja
  - **Thumbnail** – Window->Thumbnail prikazuje strukturu mapiranja

# Mapiranja

---

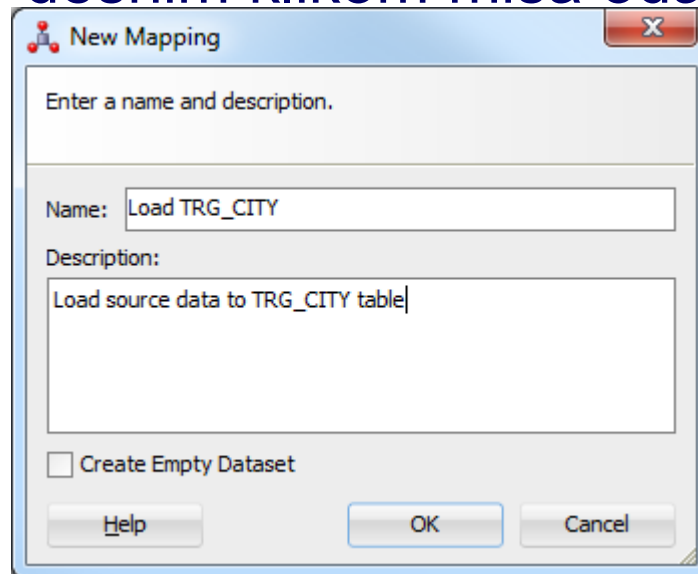
- Standardan niz koraka u postupku deklaracije mapiranja
  1. Kreirati novo mapiranje
  2. Konstrukcija mapiranja
    - U **Logical** sekciji kreirati komponente mapiranja – tabele sa atributima, definisati pravila transformacije izvornih u odredišne attribute
    - U **Physical** sekciji definisati gde (staging area) i kako (KM) se proces integracije obavlja
    - Pri konstrukciji mapiranja sve komponente se dodaju **Drag and Drop** tehnikom u **Mapping Diagram**
  3. Pokrenuti mapiranje



# Mapiranja

- Kreiranje mapiranja

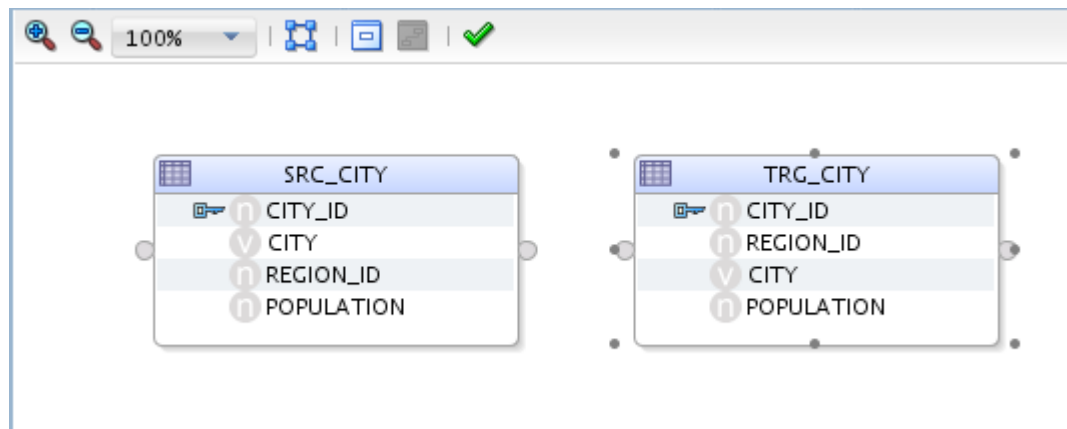
1. U **Designer** panelu, u **Project** sekciji, u okviru podrazumevanog direktorijuma kreiranog projekta, desnim klikom miša odabrati **New Mapping** opciju



2. Uneti naziv i kratak opis mapiranja, i da li želimo da kreirano mapiranje sadrži praznu tabelu
3. Sačuvati načinjene izmene

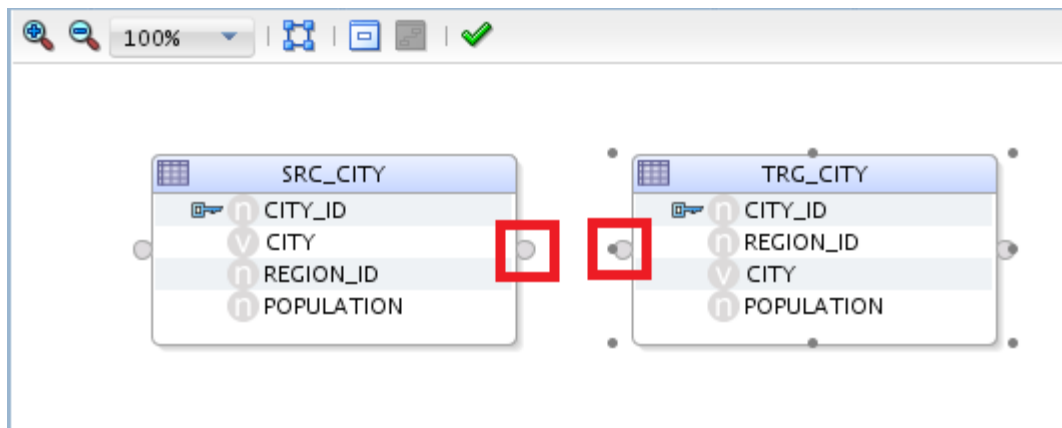
# Mapiranja

- Povezivanje komponenti mapiranja
  - Osnovno povezivanje izvornih i odredišnih tabela
    1. Prevući tabele iz izvornog i odredišnog modela



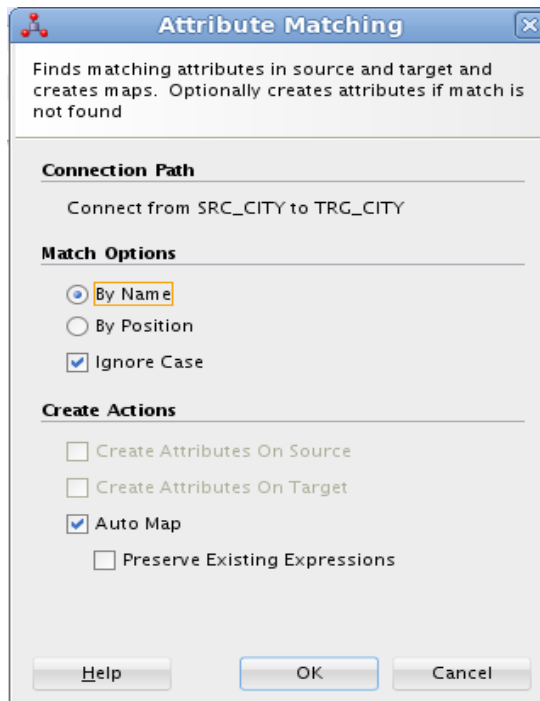
# Mapiranja

- Povezivanje komponenti mapiranja
  - Osnovno povezivanje izvornih i odredišnih tabela
    1. Povezati tabele povlačenjem veze između konektora (obeleženi crvenom kvadratima)
    2. Povezati tabele povlačenjem veze između konektora (obeleženi crvenom kvadratima)
      - Ukoliko posle povezivanja tabela, ima potrebe za dodavanjem novih elemenata mapiranja između povezanih elemenata, brisanjem veze moguće je sačuvati postojeće stanje povezanih elemenata, tako da se nakon dodavanja novih elemenata ne mora ponovo specificirati način mapiranja izvornih i odredišnih komponenti



# Mapiranja

- Povezivanje komponenti mapiranja
  - Osnovno povezivanje izvornih i odredišnih tabela
    3. Odabrati način na koji se izvorni atributi mapiraju na odredišne (preko naziva ili preko pozicije u tabeli)
      - Ili povezivanjem svakog atributa pojedninačno, revlačenjem atributa iz izvorniše tabele u atribut odredišne tabele



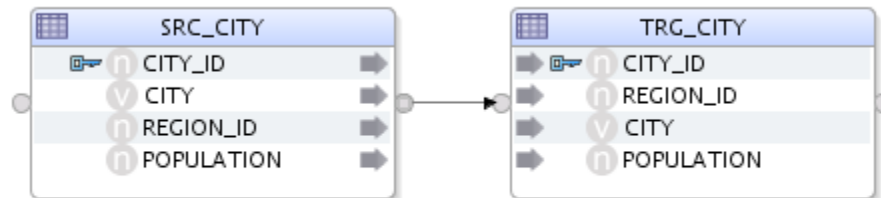
# Mapiranje


---

- Povezivanje komponenti mapiranja
  - Osnovno povezivanje izvornih i odredišnih tabela
    - **Create Actions** sekcija
      1. **Crerate Attributes On Source** – ukoliko ne postoje atributi u izvornoj tabeli, biće kreirani na osnovu atributa u odredišnoj
      2. **Create Attributes On Target** – ukoliko ne postoje atributi u odredišnoj tabeli, biće kreirani na osnovu atributa u izvornoj
      3. **Auto Map** – ukoliko postoje atributi i u izvornoj i u odredišnoj tabeli, biće povezani automatski, na osnovu zadatog kriterijuma
      4. **Preserve Existing Expressions** – u slučaju da je tabela bila povezana, pa veza izbrisana, da li pri novom povezivanju treba sačuvati podatke o povezivanju iz prethodnog povezivanja

# Mapiranja

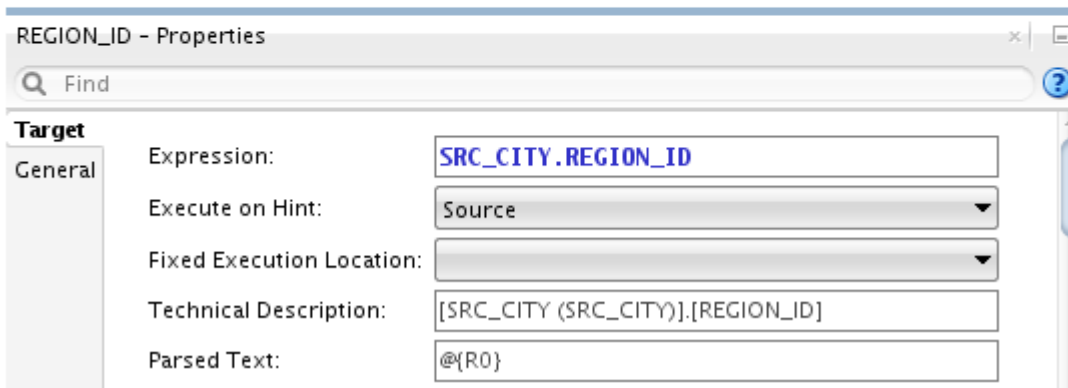
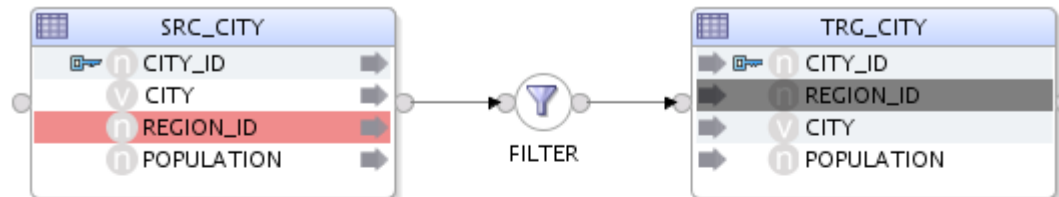
- Povezivanje komponenti mapiranja
    - Osnovno povezivanje izvornih i odredišnih tabela
4. Sačuvati načinjene izmene



- Moguće je povezivanje pojedinačnih atributa izvornih tabela sa atributima odredišnih tabela, prevlačenjem izvornih atributa na željene odredišne attribute
- Proveriti validnost kreiranog mapiranja primenom **Validate Mapping** funkcije 

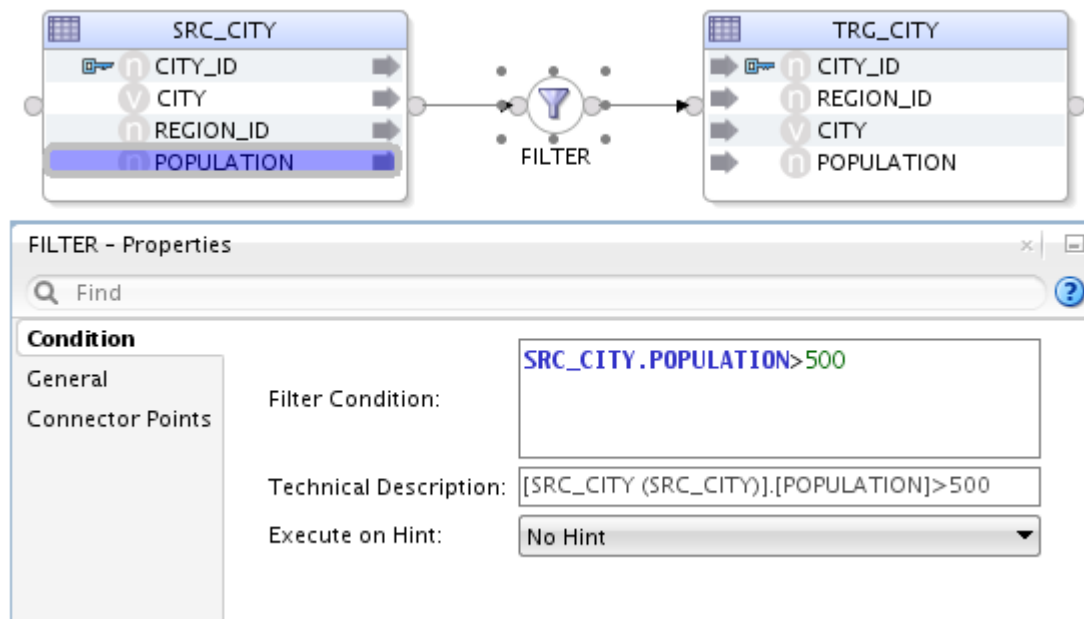
# Mapiranja

- Izrazi i logički uslovi
  - Koriste se za mapiranje pojedinačnih atributa izvornih i odredišnih tabela
  - Svaki odredišni atribut ima definisan izraz koji opisuje način na koji se odgovarajući izvorni atributi mapiraju na njega preko veze između povezanih tabela



# Mapiranja

- Izrazi i logički uslovi
  - **Filter, Join i Lookup** komponente koriste SQL izraze koji odgovaraju tipu atributa koji učestvuju u izrazu



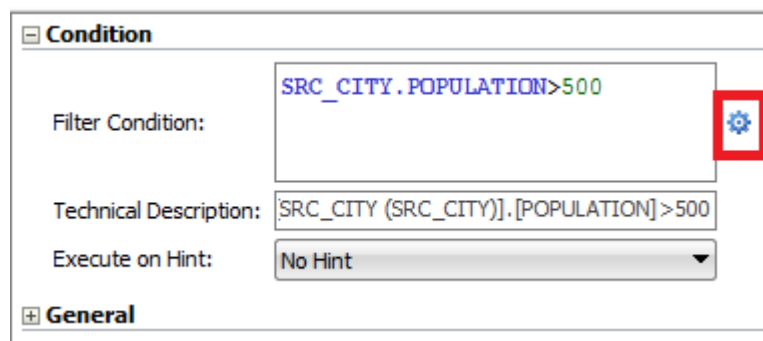
- **Expression Editor**



# Mapiranja

- **Expression Editor**

- Većina komponenti koje se koriste pri kreiranju mapiranja predstavljaju izraze koji se izvršavaju nad izvornim podacima da bi se dobili određeni podaci
- Za kreiranje kompleksnih izraza, kao i pri korišćenju ODI Mapping komponenti korisno je koristiti **Expression Editor** alat



The screenshot shows a software interface for editing expressions. It is divided into two main sections: 'Condition' and 'General'. The 'Condition' section is currently expanded and contains three fields:

- Filter Condition:** A text input field containing the expression `SRC_CITY.POPULATION>500`. To the right of this field is a red square icon containing a gear symbol, indicating a configuration or edit function.
- Technical Description:** A text input field containing the expression `[SRC_CITY (SRC_CITY)].[POPULATION]>500`.
- Execute on Hint:** A dropdown menu with the selected option being 'No Hint'.

The 'General' section is collapsed and located below the 'Condition' section.

# Mapiranje

## • Expression Editor

The screenshot shows the "Edit Filter Condition for 'FILTER'" dialog box. The main text area contains the SQL expression `SRC_CITY.POPULATION>500`. The left sidebar shows the "Attributes" tree with `SRC_CITY` expanded to show `CITY_ID`, `CITY`, `REGION_ID`, and `POPULATION`. The bottom section contains "Functions" (Aggregate, Keywords, Math, Operators) and "ODI objects" (OdiRef Functions). The bottom of the dialog has "Help", "Apply", "OK", and "Cancel" buttons.

Attributes:

- SRC\_CITY
  - CITY\_ID
  - CITY
  - REGION\_ID
  - POPULATION

SQL (Generic SQL) Highlight: the whole expression

SRC\_CITY.POPULATION>500

Functions:

- Aggregate
- Keywords
- Math
- Operators

ODI objects:

- OdiRef Functions

Help Apply OK Cancel

# Mapiranja

---

- Komponente mapiranja
  - **Projector Components**
    - Koriste attribute koji predstavljaju podatke u procesu mapiranja
    - Sadrže svoje sopstvene attribute koji potiču od atributa izvornih tabela
  - **Selector Components**
    - Koriste attribute koji predstavljaju podatke u procesu mapiranja
    - Ne kreiraju sopstvene attribute, već direktno koriste attribute izvornih tabela

# Mapiranje

---

- **Projector Components**

- **Source and Target Datastores**

- Pri kreiranju mapiranja, izvorne i odredišne tabele koje učestvuju u mapiranju, dodaju se prevlačenjem iz modela kojima propadaju, iz **Designer Navigator** panela
    - Da bi videli podatke u tabeli koja učestvuje u kreiranju mapiranja, desnim klikom na tabelu odabрати **Data** opciju
    - Tabela predstavlja izvornu tabelu ukoliko ima makar jednu vezu koja vodi prema drugim komponentama i ni jednu vezu koja vodi prema njoj
    - Tabela predstavlja odredišnu tabelu ukoliko ima makar jednu vezu koja vodi prema njoj i ni jednu vezu koja vodi od nje prema drugim komponentama koje učestvuju u kreiranju mapiranja

# Mapiranja

## • Projector Components

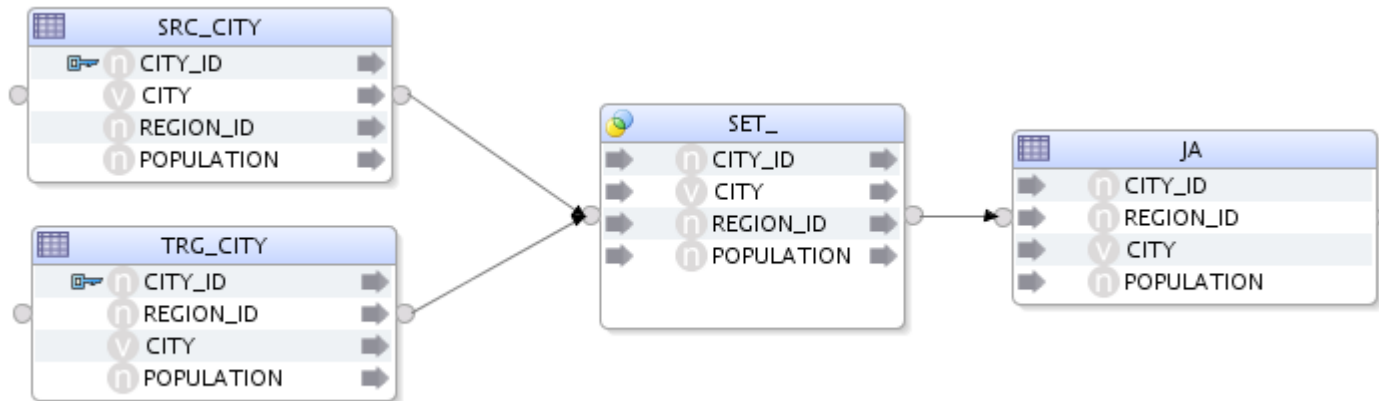
### – Sets

- Primena operacija UNION, INTERSECT, EXCEPT i MINUS na pridružene parove atributa ulaznih tabela (minimalno dve)
- Koraci kreiranja **Set** operacije
  1. Prevući **Set** komponentu mapiranja u radni prostor **Expression Editor**
  2. Prevući ulazne tabele na čije atribute treba primeniti operacije
  3. Povezati tabele sa ulaznim konektorom **Set** komponente
    1. Povezivanjem korespodentnih atributa ulaznih tabela preko imena ili pozicije u tabeli (**Attribute Matching** dijalog)
    2. Povezivanjem jedne od ulaznih tabela sa ulazom **Set** komponente i prevlačenjem korespodentnih atributa ostalih tabela
    3. Povezivanjem izlaza **Set** komponente sa odredišnom tabelom

# Mapiranja

- **Projector Components**

- **Sets**



**Attributes**

Attributes:  + ✖ 📄

Target	Expression for : INPUT1 (SRC_...	Expression for : INPUT2 (TRG_...
CITY_ID	SRC_CITY.CITY_ID	TRG_CITY.CITY_ID
CITY	SRC_CITY.CITY	TRG_CITY.CITY
REGION_ID	SRC_CITY.REGION_ID	TRG_CITY.REGION_ID
POPULATION	SRC_CITY.POPULATION	TRG_CITY.POPULATION

**General**

**Operators**

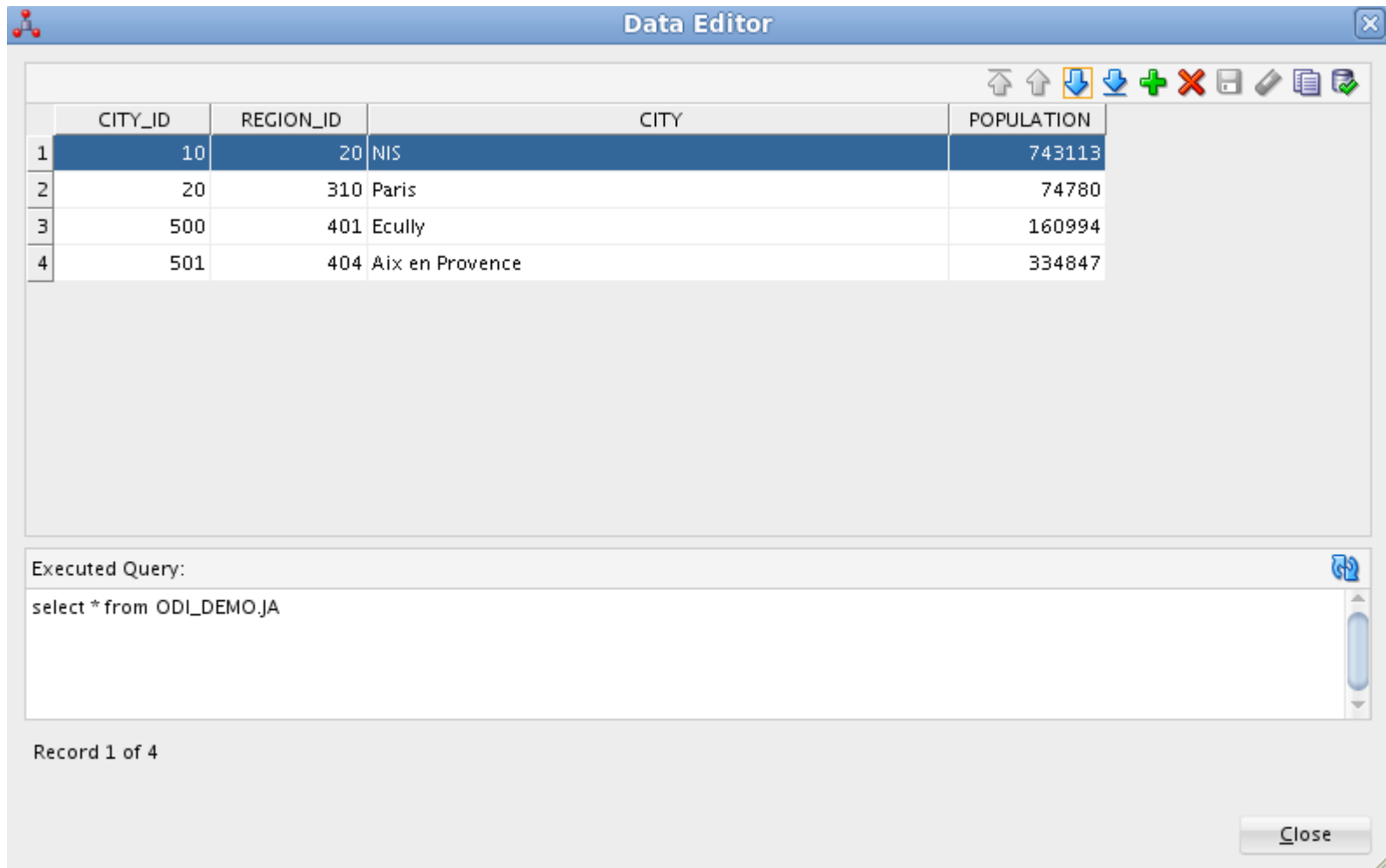
➔ Input Connector Points ➔ Output Connector Points

Input Connector Points:  + ✖ 📄

Name	Description	Bound Object	Connected From	Operator
INPUT1			SRC_CITY	
INPUT2			TRG_CITY	MINUS

# Mapiranje

- **Projector Components**
  - **Sets - rezultat**



The screenshot shows a 'Data Editor' window with a table of data. The table has four columns: CITY\_ID, REGION\_ID, CITY, and POPULATION. The first row is highlighted in blue. Below the table, there is a section for the 'Executed Query' which contains the SQL statement 'select \* from ODI\_DEMO.JA'. At the bottom left, it says 'Record 1 of 4' and at the bottom right, there is a 'Close' button.

	CITY_ID	REGION_ID	CITY	POPULATION
1	10	20	NIS	743113
2	20	310	Paris	74780
3	500	401	Ecully	160994
4	501	404	Aix en Provence	334847

Executed Query:  
select \* from ODI\_DEMO.JA

Record 1 of 4

Close

# Mapiranja

- **Projector Components**

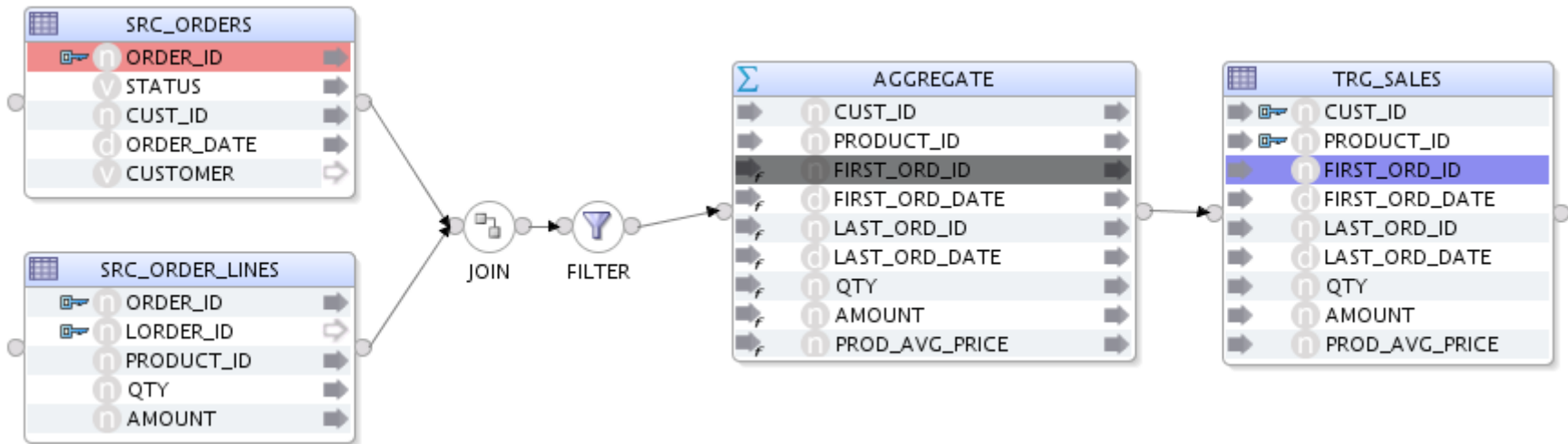
- **Aggregates**

- Služe za primenu agregacionih funkcija (average, count, max, sum..) nad atributima tabela
- Postupak kreiranja i povezivanja isti kao kod **Set** komponente
- Svaki od atributa **Aggregates** komponente kome nije pridružena ni jedna od agregacionih funkcija se koristi u **group by** uslovu.
- Alat takve attribute prepoznaje i automatski kreira **group by** uslov
  - **Is Group By>Auto**, u okviru **Properties>Attributes** sekcije **Aggregates** komponente
    - » Moguce je još postaviti i **Yes** ili **No** vrednosti ukoliko nije izabrana vrednost **Auto**, čime se naznačava da li je dati atribut deo group by kazuze, bez obzira na to da li mu je pridružena neka agregaciona funkcija ili ne
  - Maunuelno zadavanje uslova grupisanja
    - » **Properties>General>Manual Group By Clause**



# Mapiranje

- Projector Components
  - Aggregates



**Target**

Expression:

Execute on Hint:

Fixed Execution Location:

**General**

Name:

Description:

Data Type:

Size:

Scale:

# Mapiranje

---

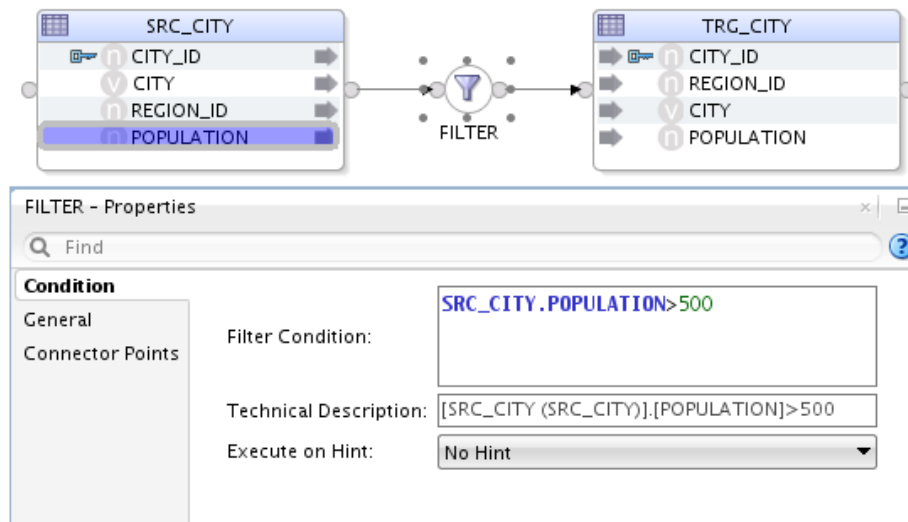
- **Projector Components**
  - **Sorts**
    - Orger by SQL klauzula
  - **Distincts**
    - Distinct SQL klauzula

# Mapiranja

## • Selector Components

### • Filters

- Filter vrši selekciju podataka koji zadovoljavaju zadati logički uslov nad nekom tabelom
- Sintaksa kojom se zadaje uslov odgovara **SQL where** klauzuli
- Filter može biti zadat u okviru tabele koja učestvuje u mapiranju (**Filters** sekcija u **Models** panelu) ili u okviru mapiranja preko **Filter** komponente mapiranja
- U filter prevući one atribute koji će učestvovati u zadavanju uslova

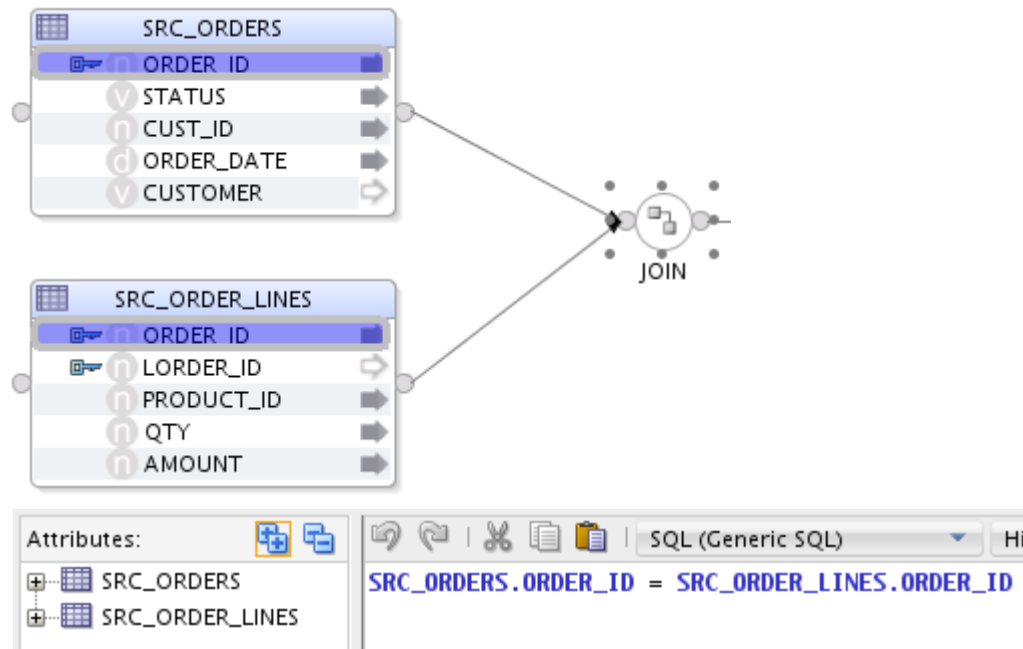


# Mapiranja

- **Selector Components**

- **Joins**

- » Join (spoj) se najčešće koristi za spajanje tabela po istim vrednostima zajedničkih obeležja
- » Prevući attribute koji učestvuju u formiranju uslova spajanja
- » U okviru **Expression** editora popuniti **Join Condition**



# Mapiranja

## • Selector Components

### • Joins

- U okviru **Join Properties** panela, u **Condition** sekciji, moguće je odabrati tip spajanja
  - » **Cross** (Dekartov proizvod)
  - » **Natural**
  - » **Right Outer**
  - » **Lef Outer**
  - » **Full Outer** – odabirom i **Right Outer** i **Lef Outer**
  - » **Inner** – ne odabiranjem ni jedne od opcija

Join Type:

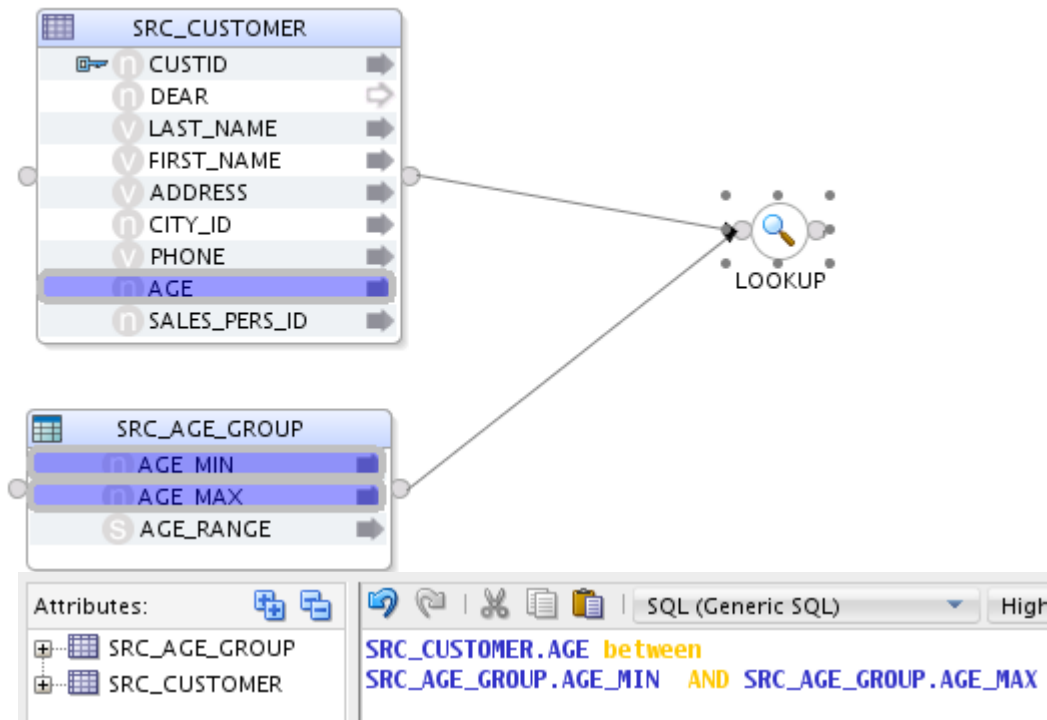
The screenshot shows a 'Join Type' selection interface. It features five radio button options: 'Cross', 'Natural', 'Left Outer Join', 'Right Outer Join', and 'Full Outer Join'. The 'Left Outer Join' option is selected, indicated by a checked checkbox. Lines connect the text 'SRC\_ORDERS(SRC\_ORDERS)' to the 'Left Outer Join' option and 'SRC\_ORDER\_LINES(SRC\_ORDER\_LINES)' to the 'Right Outer Join' option. Below the options, a descriptive text reads: 'All rows of SRC\_ORDERS(SRC\_ORDERS) including the rows unpaired with SRC\_ORDER\_LINES(SRC\_ORDER\_LINES) rows'.

# Mapiranja

## • Selector Components

### • Lookups

- » Uslov spajanja se ne zadaje preko istih vrednosti zajedničkih obeležja
- » Prevući attribute koji učestvuju u formiranju uslova spajanja
- » U okviru **Expression** editora popuniti **Join Condition**



# Mapiranja

---

- **Selector Components**

- **Expressions**

- » Koristi se najčešće kao međukomponenta za izračunavanje vrednosti atributa kojima se popunjava više odredišnih tabela
    - » Odgovara ponašanju SQL Select klauzule
    - » Pored atributa, izvorišne ili odredišne tabele koji će se koristiti u okviru Expression komponente, obično se dodaju i dodatni atributi koji će čuvati rezultate među-izračunavanja, koji će se koristiti u više odredišnih tabela

# Mapiranja

---

- **Physical Design**

- Prikazuje način izvršavanja mapiranja (runtime)
- Omogućava specifikaciju koja govori koji delovi mapiranja će se izvršavati na kom serveru, prilikom realizacija procesa mapiranja izvorišnih na određene podatke (**Execution Units**)
- Predstavlja fizičku specifikaciju komponenti definisanih na logičkom nivou

- **Deployment specification**

- Ceo Physical diagram predstavlja jedan Deployment specification



# Mapiranja

## • Physical Mapping Diagram

### – Execution Groups (označene žutim kvadratima)

- Predstavljaju grupe elemenata mapiranja koje se izvršavaju istovremeno, a odnose se na izvorišne ili odredišne podatke

- **Source Execution Group**

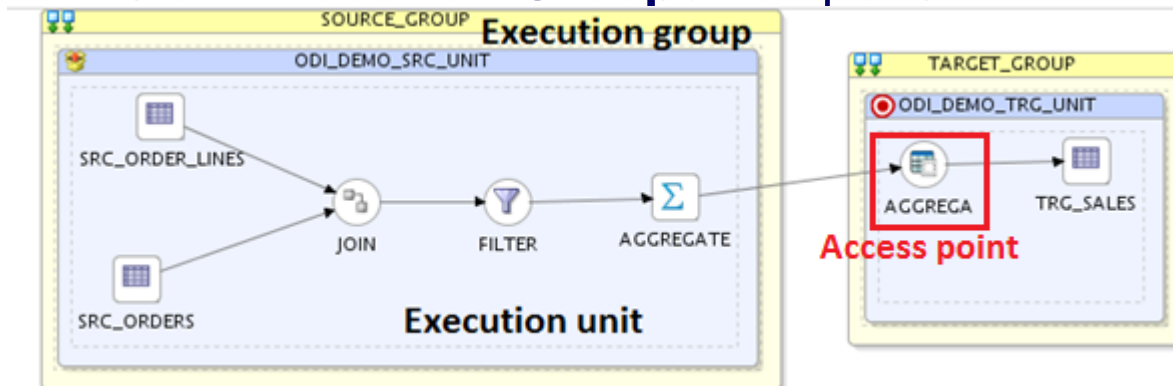
- Izvorne tabele koje učestvuju u mapiranju

- **Target Execution Group**

- Odredišne tabele koje učestvuju u mapiranju

### – Execution Units (označene plavim kvadratima)

- Delovi Execution Groups komponenti



# Mapiranja

---

- **Access Point**

- Pristupna tačka, kreirana na odredišnoj **Execution Unit** sekciji, tačka povezivanja izvorišnih i odredišnih elemenata mapiranja

- **Knowledge Modules**

- Definišu način prenosa podataka i kontrole ograničenja između izvorne i odredišne **Execution Unit** sekcije
- Izvršavaju predefinisani kod koji se sastoji iz metoda čiji se parametri eksplicitno popunjavaju na osnovu odabranih opcija sa korisničkog interfejsa

# Mapiranje

---

- **Knowledge Modules**

- **Loading Knowledge Module (LKM)**

- Bira se u okviru **Access Point** sekcije na odredišnoj **Execution Unit** sekciji
    - Definiše način na koji se podaci prenose u **staging area** koji može biti definisan na odredišnom serveru ili na posebnoj server instanci

- **Integration Knowledge Module (IKM) and Check Knowledge Modules (CKM)**

- Biraju se u okviru odredišne tabele u **Execution Unit** sekciji
    - Definišu način integracije podataka i provere ograničenja u okviru odredišnih skladišta podataka

# Mapiranja

---

- **Reusable Mapping**

- U situacijama kada se u više različitih mapiranja koriste iste komponente mapiranja
- Zajedničke komponente izdvojiti u posebno mapiranje - **Reusable Mapping**
- Tako specificirano mapiranje je moguće koristiti u okviru drugih mapiranja kao i svaku drugu komponentu mapiranja (filter, join, distinct...)

# Mapiranje

---

- **Reusable Mapping**

- Dva načina kreiranja

- Kao globalni element

- U okviru **Designer>Global Objects>Global Reusable Mappings**

- Kao element projekta

- U okviru **Designer>Projects>Reusable Mappings**

1. Desnim klikom odabрати **New Reusable Mapping**

2. Primeniti isti postupak kreiranja kao i kod regularnih mapiranja

# Mapiranja

---

- **Flow Control i Static Control**

- U okviru kreiranog mapiranja moguća su dva načina kontrole specificiranih ograničenja
  - **Flow Control**
    - Provera ograničenja nad podacima pre njihovog upisivanja u određeno skladište podataka
  - **Static Control**
    - Provera ograničenja nad podacima nakon njihovog upisivanja u određeno skladište podataka
- Odabir jednog od načina kontrole ograničenja obavlja se u okviru **Physical>Integration Knowledge Module** sekcije kreiranog mapiranja (FLOW\_CONTROL ili STATIC\_CONTROL opcije)

# Mapiranje

---

- Pokretanje mapiranja
  - Desnim klikom miša na željeni paket odabrati **Run** opciju
  - Selektovati željeno mapiranje i odabrati **Run** opciju iz ODI Toolbar sekcije
  - Selektovati željeni paket pa iz menija ODI Menu sekcije odabrati **Run>Run**
  - Rezultat izvršenja koraka nakon izvršenja paketa pogledati u okviru **Operator Navigator** panela

# Sadržaj

---

- Kreiranja integracionog projekta
- Kreiranje integracionog projekta
- Paketi
- Mapiranja
- Realizacija ECTL procesa



# Realizacija ECTL procesa

- Orders Application**

- Bazirana na Oracle RDBMS tehnologiji i sadrži šest tabela

SRC_REGION			
<u>REGION_ID</u>	NUMERIC(10)	<pk>	not null
REGION	VARCHAR(50)		null
COUNTRY_ID	NUMERIC(10)		null
COUNTRY	VARCHAR(50)		null

SRC_CITY			
<u>CITY_ID</u>	NUMERIC(10)	<pk>	not null
CITY	VARCHAR(50)		null
REGION_ID	NUMERIC(10)		null
POPULATION	NUMERIC(10)		null

SRC_ORDER_LINES			
<u>ORDER_ID</u>	NUMERIC(10)	<pk>	not null
<u>LORDER_ID</u>	NUMERIC(10)	<pk>	not null
PRODUCT_ID	NUMERIC(10)		null
QTY	NUMERIC(10)		null
AMOUNT	NUMERIC(10,2)		null

SRC_CUSTOMER			
<u>CUSTID</u>	NUMERIC(10)	<pk>	not null
DEAR	NUMERIC(1)		null
LAST_NAME	VARCHAR(50)		null
FIRST_NAME	VARCHAR(50)		null
ADDRESS	VARCHAR(100)		null
CITY_ID	NUMERIC(10)		null
PHONE	VARCHAR(50)		null
AGE	NUMERIC(3)		null
SALES_PERS_ID	NUMERIC(10)		null

SRC_ORDERS			
<u>ORDER_ID</u>	NUMERIC(10)	<pk>	not null
STATUS	VARCHAR(3)		null
CUST_ID	NUMERIC(10)		null
ORDER_DATE	DATE		null
CUSTOMER	VARCHAR(35)		null

SRC_PRODUCT			
<u>PRODUCT_ID</u>	NUMERIC(10)	<pk>	not null
PRODUCT	VARCHAR(50)		null
PRICE	NUMERIC(10,2)		null
FAMILY_NAME	VARCHAR(50)		null

# Realizacija ECTL procesa

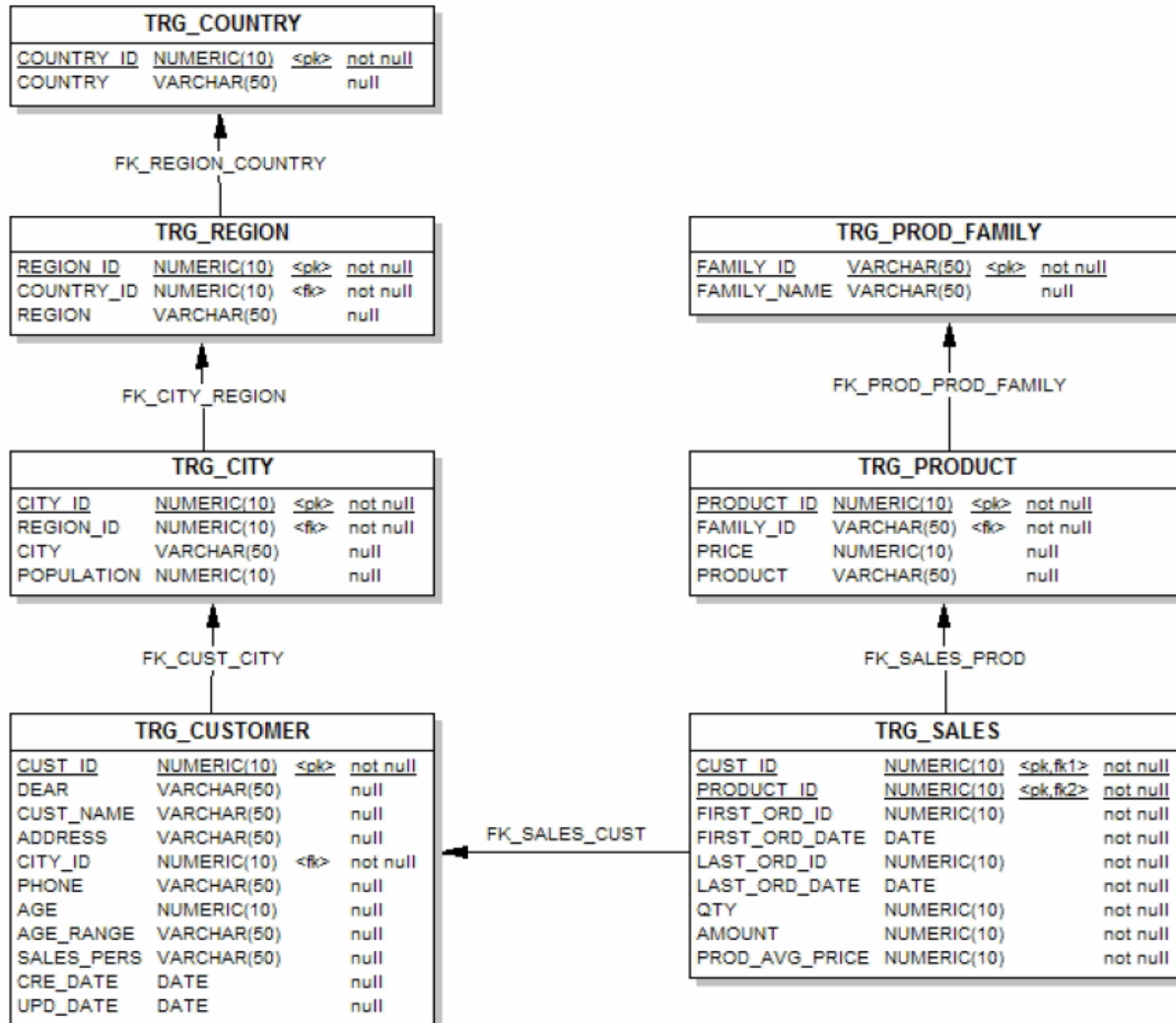
- Parameters

SRC_SALES_PERSON			
<u>SALES_PERSON ID</u>	<u>NUMERIC(10)</u>	<u>&lt;pk&gt;</u>	<u>not null</u>
FIRST_NAME	VARCHAR(50)		null
LAST_NAME	VARCHAR(50)		null
HIRE_DATE	DATE		null

SRC_AGE_GROUP			
<u>AGE_MIN</u>	<u>NUMERIC(3)</u>	<u>&lt;pk&gt;</u>	<u>not null</u>
<u>AGE_MAX</u>	<u>NUMERIC(3)</u>	<u>&lt;pk&gt;</u>	<u>not null</u>
AGE_RANGE	VARCHAR(50)		null

# Realizacija ECTL procesa

## • Sales Administration



# Realizacija ECTL procesa

---

- **Cilj**

- Praćenje podataka o prodaji prema proizvodima i kupcima
- Činjenična dimenzija: **Sales**
  - Za svakog korisnika i naručeni proizvod
    - ID prve prужbe i datum
    - ID poslednje porudžbe i datum
    - Kupljena količina
    - Suma potrošenog novca
    - Srednja vrednost potrošenog novca

# Realizacija ECTL procesa

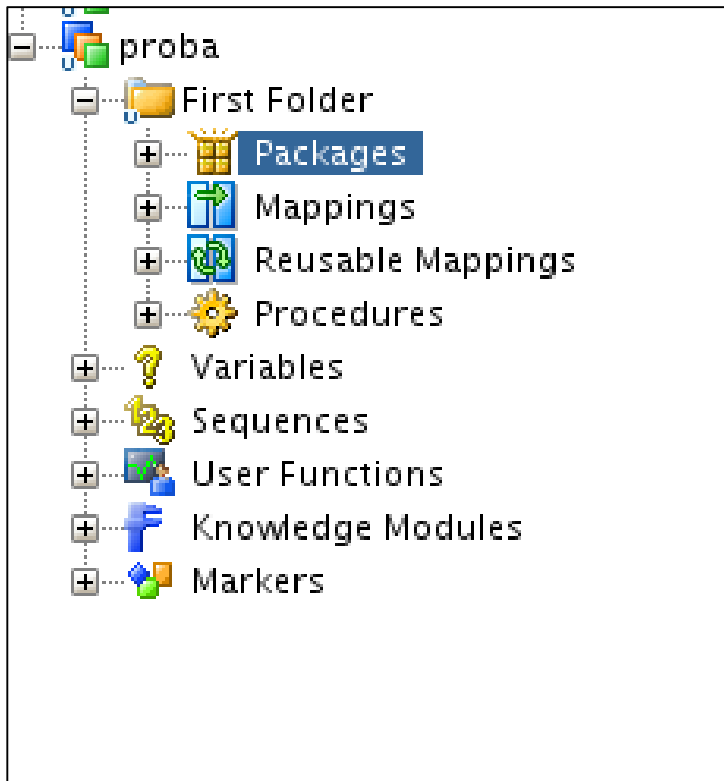
---

- **Zadaci**

- Kreirati mapiranja koja će obezbediti automatizaciju procesa izdvajanja podataka iz izvorne baze podataka, kao i operacije agregacije i integracije podataka u DW bazu podataka
- Kreirati pakete kao skupove mapiranja i konfigurisati način izvršavanja kreiranog mapiranja u okviru paketa
- Izvršiti kreirani paket i analizirati rezultate izvršavanja kreiranih mapiranja u okvir alata ODI
- Implementirati strategije izdvajanja, prečišćavanja, analize, agregacije i punjenja podataka u DW bazu podataka

# Realizacija ECTL procesa

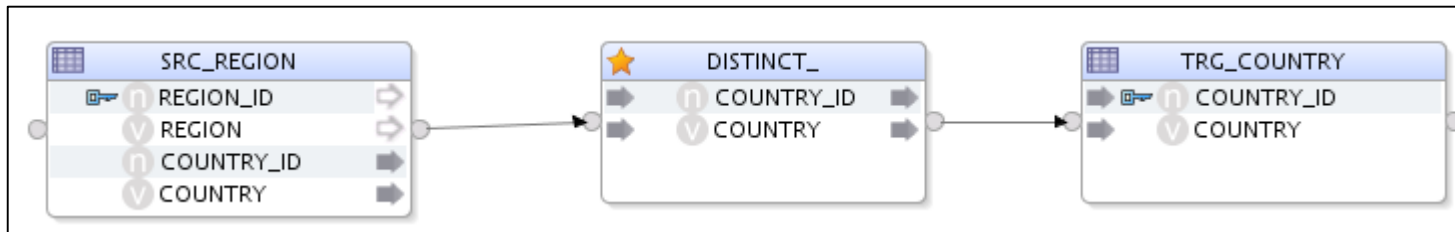
- **Napraviti novi projekat u okviru Designer panela**



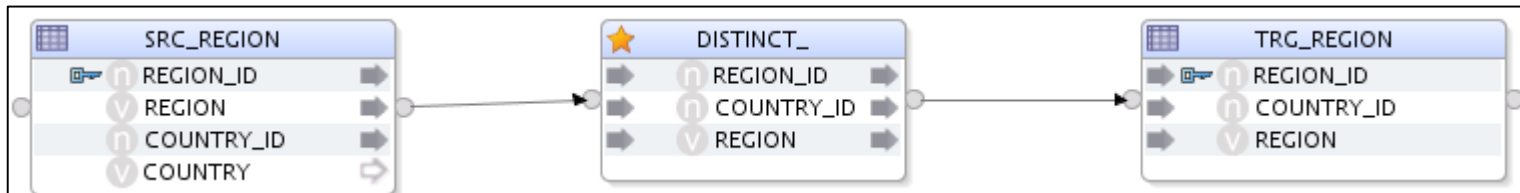
- **Infrastruktura?**

# Realizacija ECTL procesa

- Kreirati sledeća mapiranja
  - Load\_country

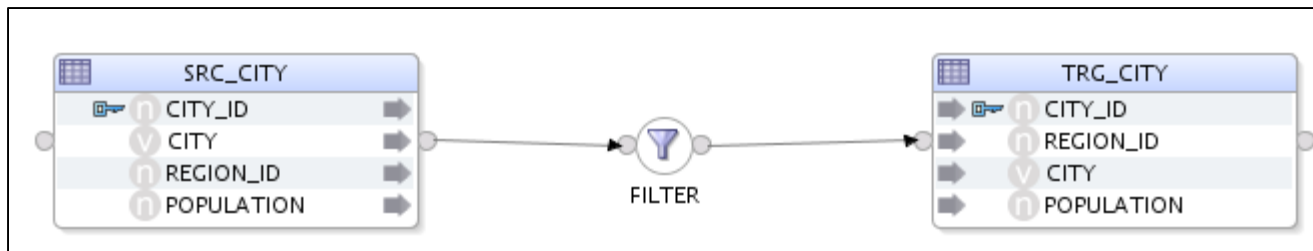


- Load\_region



# Realizacija ECTL procesa

- Kreirati sledeća mapiranja
  - Load\_city

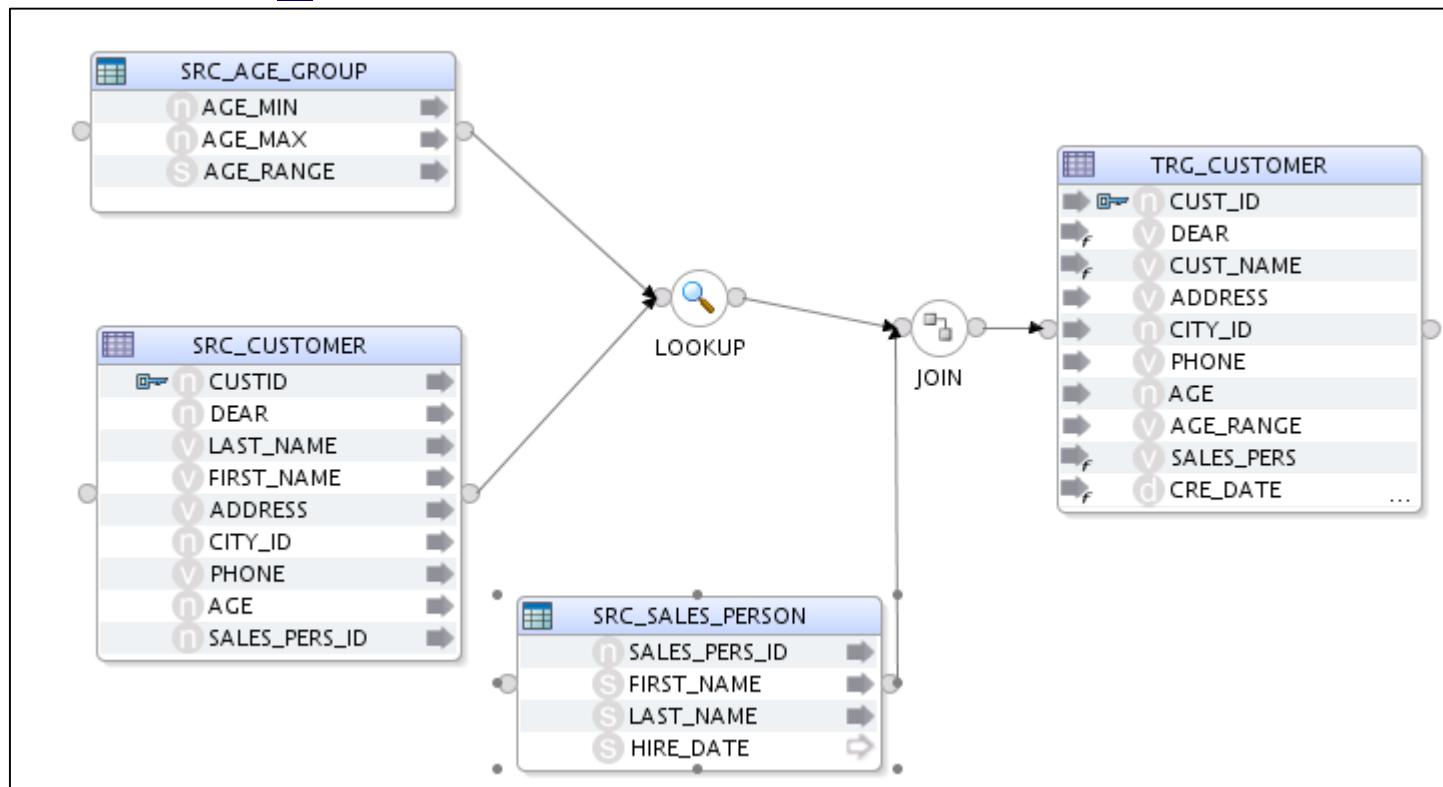


<b>Condition</b>	
Filter Condition:	<code>SRC_CITY.POPULATION&gt;500</code>
Technical Description:	<code>[SRC_CITY (SRC_CITY)].[POPULATION]&gt;500</code>
Execute on Hint:	No Hint
<b>General</b>	
<b>Connector Points</b>	



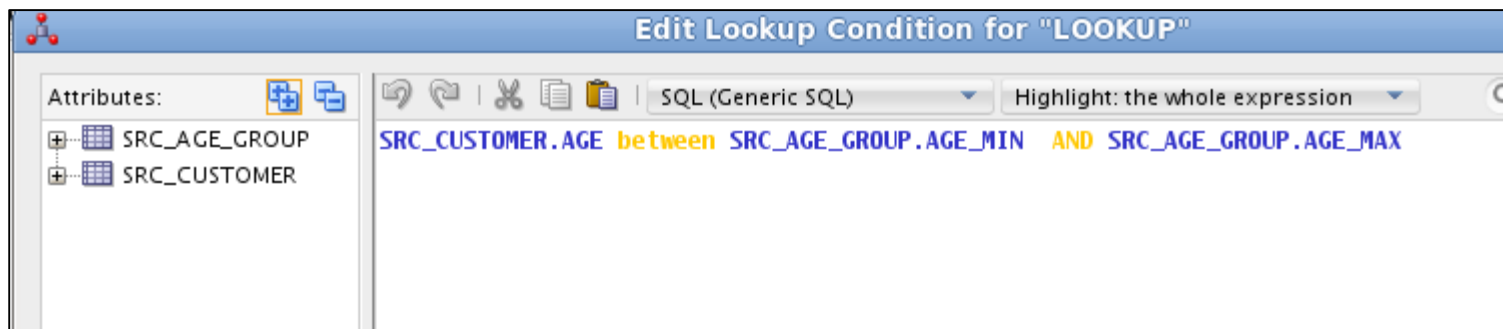
# Realizacija ECTL procesa

- Kreirati sledeća mapiranja
  - Load\_customer



# Realizacija ECTL procesa

- **Kreirati sledeća mapiranja**
  - Load\_customer



# Realizacija ECTL procesa

- Kreirati sledeća mapiranja
  - Load\_customer

Messages - Log JOIN - Properties

Find

**Condition**

Join Condition: SRC\_CUSTOMER.SALES\_PERS\_ID = SRC\_SALES\_PERSON.SALES\_PERS\_ID

Join Type: LOOKUP(LOOKUP) Cross Natural SRC\_SALES\_PERSON(SRC\_S...  
Inner Join

All rows paired by the join condition between LOOKUP(LOOKUP) and SRC\_SALES\_PERSON(SRC\_SALES\_PERSON)

Technical Description: [SRC\_CUSTOMER (SRC\_CUSTOMER)].[SALES\_PERS\_ID] = [SRC\_SALES\_PERSON (SRC\_S...

Generate ANSI Syntax:

Join Order:  User Defined:

Execute on Hint: No Hint

**General**

# Realizacija ECTL procesa

- **Kreirati sledeća mapiranja**
  - Load\_customer

Messages - Log JOIN - Properties x

Find ?

**Condition**

Join Condition: SRC\_CUSTOMER.SALES\_PERS\_ID = SRC\_SALES\_PERSON.SALES\_PERS\_ID

Join Type: LOOKUP(LOOKUP) Cross Natural SRC\_SALES\_PERSON(SRC\_S...  
Inner Join

All rows paired by the join condition between LOOKUP(LOOKUP) and SRC\_SALES\_PERSON(SRC\_SALES\_PERSON)

Technical Description: [SRC\_CUSTOMER (SRC\_CUSTOMER)].[SALES\_PERS\_ID] = [SRC\_SALES\_PERSON (SRC\_S...

Generate ANSI Syntax:

Join Order:  User Defined:

Execute on Hint: No Hint

**General**

# Realizacija ECTL procesa

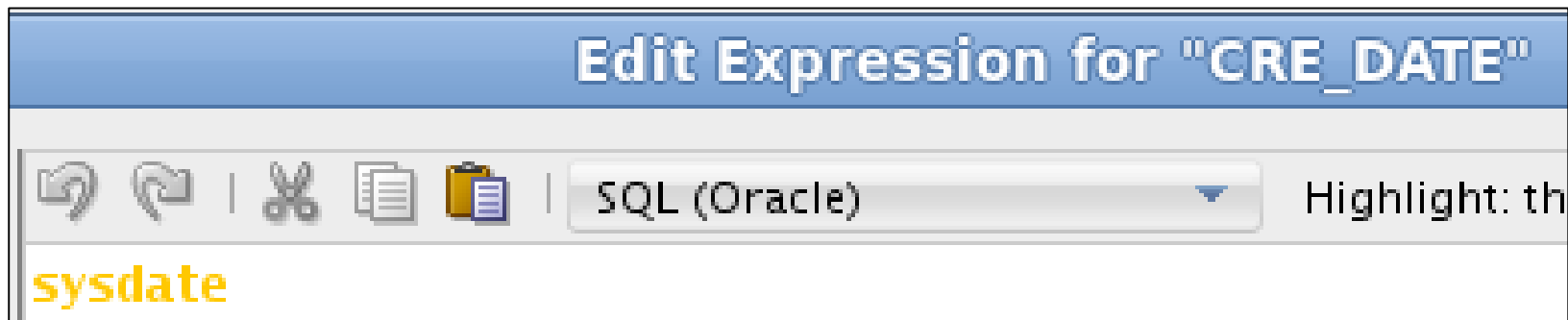
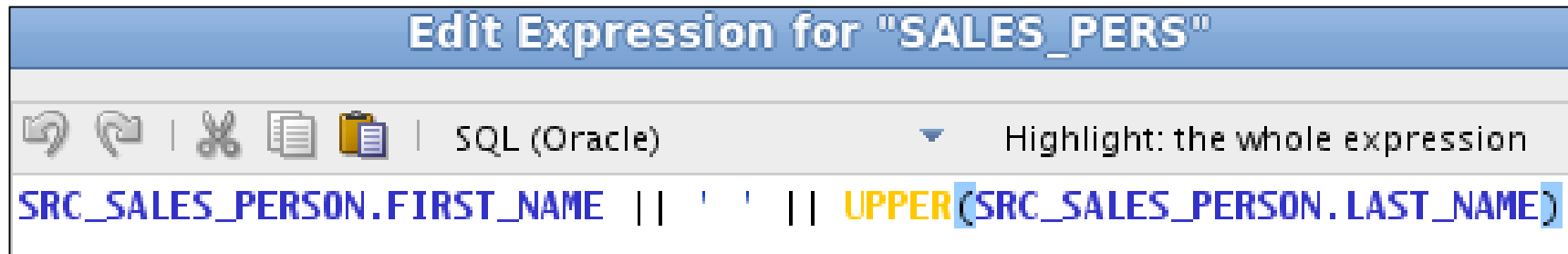
- Kreirati sledeća mapiranja
  - Load customer

```
Edit Expression for "DEAR"  
SQL (Oracle)  
CASE  
WHEN SRC_CUSTOMER.DEAR = 0 THEN 'Mr'  
WHEN SRC_CUSTOMER.DEAR = 1 THEN 'Mrs'  
ELSE 'Ms'  
END
```

```
Edit Expression for "CUST_NAME"  
SQL (Oracle)  
SRC_CUSTOMER.FIRST_NAME || ' ' || UPPER(SRC_CUSTOMER.LAST_NAME)
```

# Realizacija ECTL procesa

- **Kreirati sledeća mapiranja**
  - Load\_customer



- Insert

# Realizacija ECTL procesa

- **Kreirati sledeća mapiranja**
  - Load\_product\_family



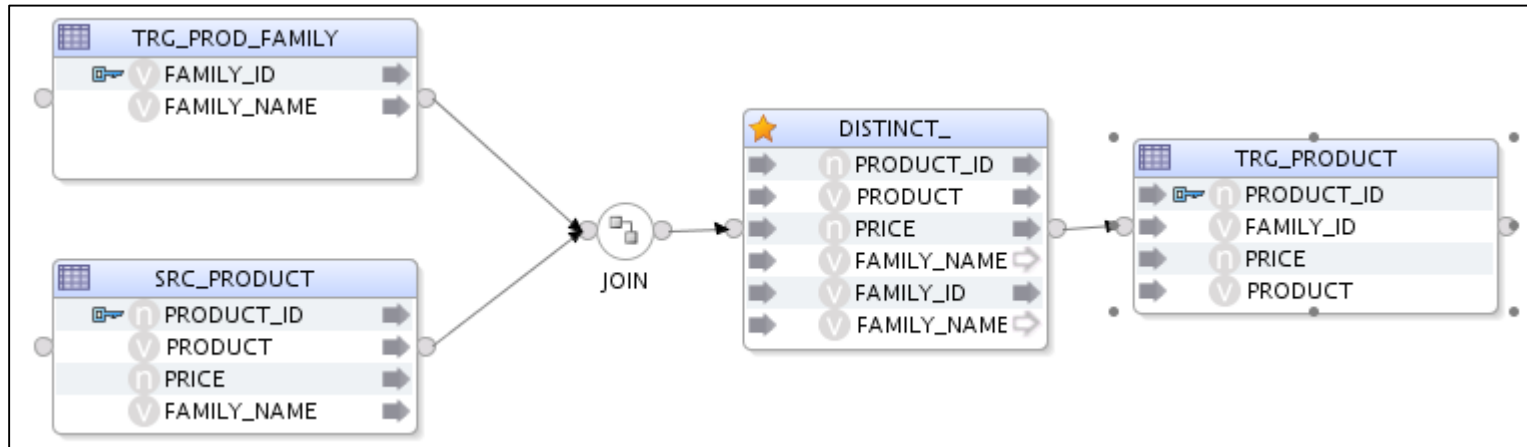
**Edit Expression for "FAMILY\_ID"**

SQL (Oracle) Highlight: the whole e

```
upper(SUBSTR(DISTINCT_.FAMILY_NAME,1,3))
```

# Realizacija ECTL procesa

- Kreirati sledeća mapiranja
  - Load\_product



**Condition**

Join Condition: `SRC_PRODUCT.FAMILY_NAME = TRG_PROD_FAMILY.FAMILY_NAME`

Join Type:
   
 Cross     Natural
   
 Inner Join

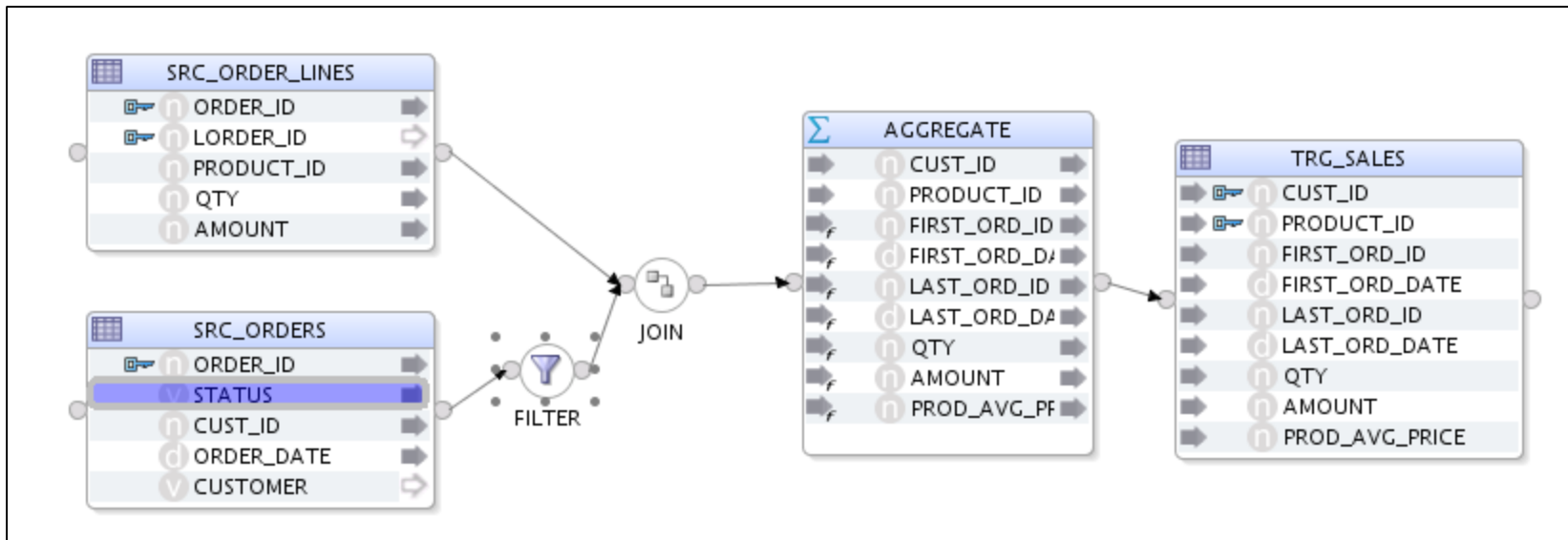
All rows paired by the join condition between SRC\_PRODUCT(SRC\_PRODUCT) and TRG\_PROD\_FAMILY(TRG\_PROD\_FAMILY)

Technical Description: `[SRC_PRODUCT (SRC_PRODUCT)].[FAMILY_NAME] = [TRG_PROD_FAMILY (TRG_PROD_FAMILY)].[FAMILY_NAME]`



# Realizacija ECTL procesa

- Kreirati sledeća mapiranja
  - Load\_sales



# Realizacija ECTL procesa

- Kreirati sledeća mapiranja
  - Load\_sales

Messages - Log FILTER - Properties x

Find

**Condition**

Filter Condition: SRC\_ORDERS.STATUS='CLO'

Messages - Log JOIN - Properties x

Find

**Condition**

Join Condition: SRC\_ORDER\_LINES.ORDER\_ID = SRC\_ORDERS.ORDER\_ID

Join Type:

Cross  Natural  Inner Join

SRC\_ORDER\_LINES(SRC\_OR... SRC\_ORDERS(SRC\_ORDERS)

# Realizacija ECTL procesa

- Kreirati sledeća mapiranja
  - Load\_sales

Messages - Log FIRST\_ORD\_ID - Properties x

Find

Target

`MIN(SRC_ORDERS.ORDER_ID)`

Messages - Log FIRST\_ORD\_DATE - Properties x

Find

Target

`MIN(SRC_ORDERS.ORDER_DATE)`

Expression:

Messages - Log LAST\_ORD\_ID - Properties x

Messages - Log

Target

`MAX(SRC_ORDERS.ORDER_ID)`

# Realizacija ECTL procesa

- **Kreirati sledeća mapiranja**
  - Load\_sales

Messages - Log    LAST\_ORD\_DATE - Properties ×

Find

Target

`MAX(SRC_ORDERS.ORDER_DATE)`

Messages - Log    QTY - Properties ×

Find

Target

`SUM(SRC_ORDER_LINES.QTY)`

Messages - Log    AMOUNT - Properties ×

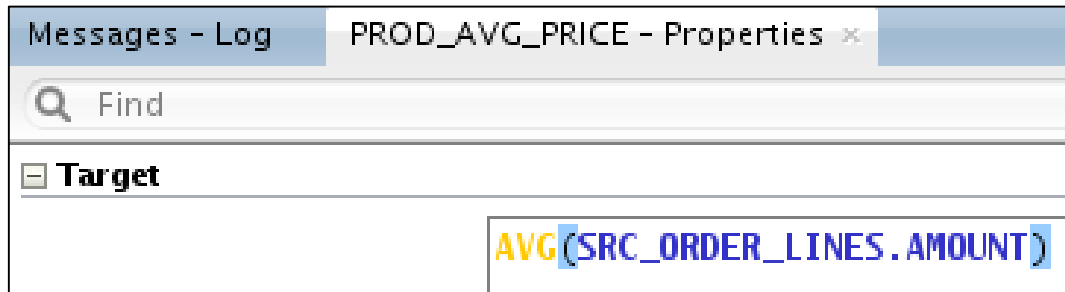
Find

Target

`SUM(SRC_ORDER_LINES.AMOUNT)`

# Realizacija ECTL procesa

- **Kreirati sledeća mapiranja**
  - Load\_sales



# Realizacija ECTL procesa

- Kreirati paket i definisati korake izvršavanja



- Pokrenuti pakete i pregledati rezultate

# Pitanja i komentari

---



# Sadržaj

---

- Kreiranje integracionog projekta
- Kreiranje integracionog projekta
- Paketi
- Mapiranje
- Realizacija ECTL procesa



Kraj prezentácie

# Sistemi skladišta podataka



## Vežba 3

---