

Laboratorijska vežba 1

Cilj vežbe: Kreiranje Data Mart-a (dimenzionalnog modela) za poslovni proces prodaje (Sales) u okviru jednog preduzeća. U okviru vežbe prikazani su svi koraci za kreiranje i popunjavanje dimenzija Star scheme na osnovu informacija koje se pribavljaju iz baze podataka, kao i definisanje i popunjavanje tabele činjenica Star scheme korišćenjem Apache NIFI alata.

Instalacija potrebnih alata

- Potrebno je da na računaru bude instalirana Java 8 ili Java 11
- Podesiti JAVA_HOME promenljivu okruženja (environment variable)

Instalacija Apache NIFI

- Preuzeti Apache NIFI sa adrese <https://nifi.apache.org/download.html>
- Raspakovati Apache NIFI na željenu lokaciju

Instalacija Oracle JDBC drajvera

Kako bi se omogućila konekcija na Oracle server potrebno je instalirati Oracle JDBC drajver.

- Preuzeti ojdbc8.jar sa adrese [Oracle JDBC link](#)
- Iskopirati JDBC drajver u folder nifi-1.19.1\lib

Apache NIFI

Apache NIFI je alat koji se koristi za automatizaciju i upravljanje protoka podataka između sistema. Pruža mogućnost kreiranja tokova podataka između različitih izvora i odredišta, rutiranje i transformaciju podataka. Apache NIFI ima Web korisnički interfejs za kreiranje tokova podataka, njihovu kontrolu i praćenje.

Osnovni koncepti Apache NIFI alata su:

- FlowFile – predstavlja svaki podatak koji korisnik unosi u NIFI radi obrade i distribucije. Svaki FlowFile se sastoji od dva dela: atributa i sadržaja. Sadržaj su učitani podaci korisnika. Dok su atributi parovi ključ-vrednost koji pružaju informacije ili kontekst o podacima.
- Procesor - NiFi komponenta koja je odgovorna za kreiranje, slanje, prijem, transformaciju, rutiranje, razdvajanje, spajanje i obradu FlowFile-ova. To je najvažniji gradivni blok koji je dostupan korisnicima NiFi-a za kreiranje tokova podataka. Procesori imaju pristup atributima i sadržaju FlowFile-ova.

Svaki procesor ima nula ili više definisanih relacija (Relationship). Ove relacije su imenovane da ukažu na rezultat obrade FlowFile-a. Nakon što procesor završi obradu FlowFile-a, on će usmeriti FlowFile u jednu od relacija. Svaka od ovih relacija može da se poveže sa drugom komponentom kako bi se definisalo gde FlowFile treba da bude prosleđen u zavisnosti od rezultata obrade.

Procesori se povezuju vezama koje funkcionišu kao redovi koji omogućavaju različitim procesima da interaguju različitim brzinama. Svaka veza se sastoji od jedne ili više relacija procesora.

Korisni linkovi za detaljnije upoznavanje sa ovim alatom:

<https://nifi.apache.org/docs/nifi-docs/html/user-guide.html>

<https://nifi.apache.org/docs.html>

<https://nifi.apache.org/docs/nifi-docs/html/overview.html>

Pokretanje aplikacije

- Otvoriti folder **/bin**
- Duplim klikom na **run-nifi.bat** fajl pokreće se NIFI
- Prilikom prvog pokretanja biće kreirani potrebni folderi i fajlovi poput: *content_repository*, *database_repository*, *flowfile_repository*, *provenance_repository*, *work* i *logs*

Podrazumevana NIFI instalacija generiše nasumično korisničko ime i lozinku i upisuje generisane vrednosti u log aplikacije koji se nalazi na lokaciji **logs/nifi-app.log**. Log fajl sadrži redove sa generisanim korisničkim imenom i generisanom lozinkom:

Generated Username [USERNAME]

Generated Password [PASSWORD]

Nakon što je NIFI uspešno pokrenut, korisničkom interfejsu može da se pristupi preko adrese <https://localhost:8443/nifi/>

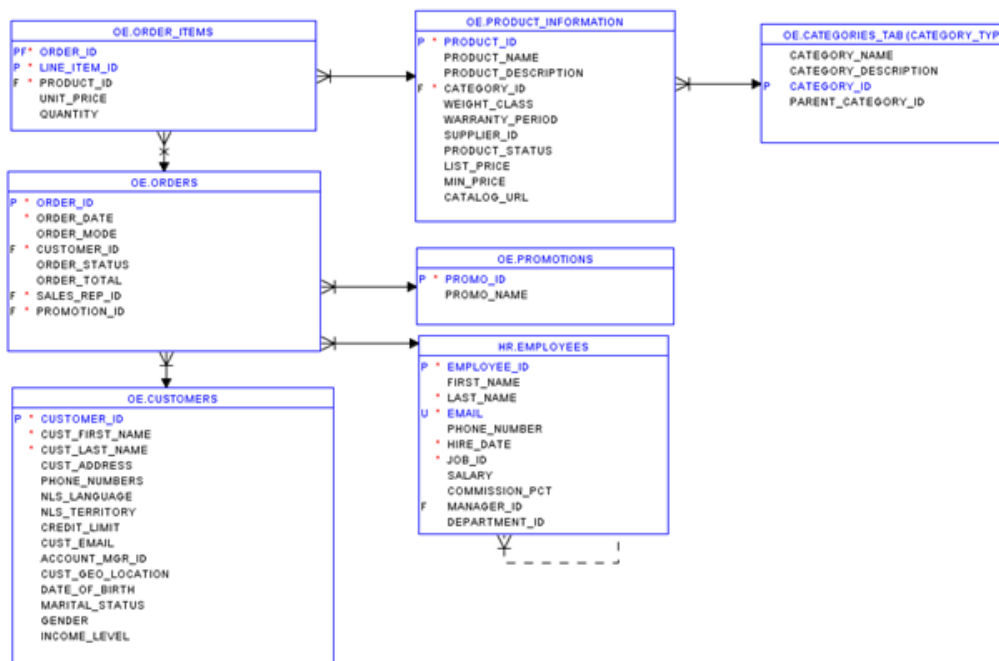
Pristup Oracle bazi podataka

Parametri	Operativna baza podataka	Data Mart
Tip konekcije:	Oracle	Oracle
Konekcija	160.99.12.92/GISLAB_PD	160.99.12.92/GISLAB_PD
Port:	1521	1521
Korisničko ime:	dwuser	S<broj indeksa>
Šifra:	ipzuser	S<broj indeksa>

Za potrebe laboratorijskih vežbi za svakog studenta je kreiran korisnički nalog na Oracle bazi podataka koja se nalazi na fakultetskom serveru 160.99.12.92. Svi korisnički nalozi su oblika **S<broj indeksa>** (Za studenta čiji je broj indeksa 11100 kreiran je korisnički nalog S11100). Inicijalno svaki korisnički nalog ima lozinku koja je ista kao i korisnički nalog (za studenta koji ima korisnički nalog S11100 inicijalna lozinka je S11100).

Vežba

Deo operativne baze podataka jednog preduzeća koji obuhvata tabele vezane za prodaju, prikazan je na slici ispod. Preduzeće prodaje proizvode iz različitih kategorija. Korisnici naručuju proizvode koji mogu ali ne moraju da budu na nekoj promociji. Za svaku narudžbinu zadužen je neki zaposleni.



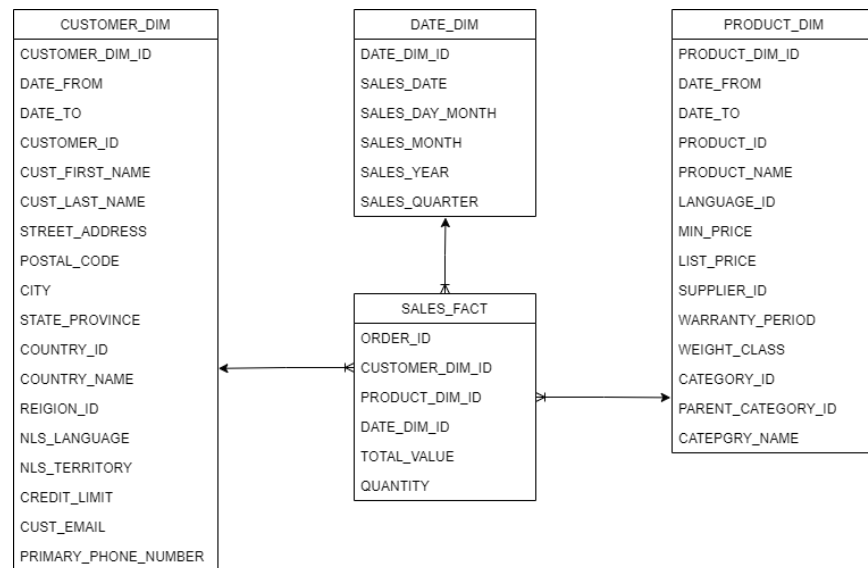
Baza podataka prodaje jednog preduzeća

U okviru laboratorijske vežbe potrebno je kreirati dimenzionalni model koji se sastoji od:

- tabele činjenica – *sales_dim*,
- i tabele dimenzija *customer_dim*, *product_dim* i *date_dim*

Dimenzionalni model (Star schema) prikazan je na slici koja se nalazi u nastavku.

Za kreiranje dimenzionalnog modela koristi se Apache NIFI alat.



Dimenzionalni model

Kreiranje baze podataka za Data Mart

1. U bazi podataka u kojoj će biti smešten Data Mart kreirati tabele dimenzija i tabelu činjenica.

1.1. Naredba za kreiranje dimenzije CUSTOMER_DIM

```
CREATE TABLE CUSTOMER_DIM
(
    CUSTOMER_DIM_ID NUMBER(*,0) PRIMARY KEY,
    DATE_FROM DATE,
    DATE_TO DATE,
    CUSTOMER_ID NUMBER(*,0),
    CUST_FIRST_NAME VARCHAR2(30 BYTE),
    CUST_LAST_NAME VARCHAR2(35 BYTE),
    STREET_ADDRESS VARCHAR2(50 BYTE),
    POSTAL_CODE NUMBER(*,0),
    CITY VARCHAR2(25 BYTE),
    STATE_PROVINCE VARCHAR2(3 BYTE),
    COUNTRY_ID VARCHAR2(2 BYTE),
    COUNTRY_NAME VARCHAR2(25 BYTE),
    REGION_ID NUMBER(*,0),
    NLS_LANGUAGE VARCHAR2(3 BYTE),
    NLS_TERRITORY VARCHAR2(11 BYTE),
    CREDIT_LIMIT NUMBER(*,0),
    CUST_EMAIL VARCHAR2(50 BYTE),
    PRIMARY_PHONE_NUMBER VARCHAR2(20 BYTE)
);
```

```
CREATE SEQUENCE CUSTOMER_DIM_S;
```

```
CREATE OR REPLACE TRIGGER CUSTOMER_DIM_ON_INSERT
  BEFORE INSERT ON CUSTOMER_DIM
  FOR EACH ROW
BEGIN
  SELECT CUSTOMER_DIM_S.NEXTVAL
  INTO :NEW.CUSTOMER_DIM_ID
  FROM DUAL;
END;
```

1.2. Naredba za kreiranje dimenzije PRODUCT_DIM

```
CREATE TABLE PRODUCT_DIM
(
  PRODUCT_DIM_ID NUMBER(*,0) PRIMARY KEY,
  DATE_FROM DATE,
  DATE_TO DATE,
  PRODUCT_ID NUMBER(*,0),
  PRODUCT_NAME VARCHAR2(35 BYTE),
  LANGUAGE_ID VARCHAR2(2 BYTE),
  MIN_PRICE NUMBER(*,0),
  LIST_PRICE NUMBER(*,0),
  SUPPLIER_ID NUMBER(*,0),
  WARRANTY_PERIOD VARCHAR2(6 BYTE),
  WEIGHT_CLASS NUMBER(*,0),
  CATEGORY_ID NUMBER(*,0),
  PARENT_CATEGORY_ID NUMBER(*,0),
  CATEGORY_NAME VARCHAR2(25 BYTE)
);

CREATE SEQUENCE PRODUCT_DIM_S;

CREATE OR REPLACE TRIGGER PRODUCT_DIM_ON_INSERT
  BEFORE INSERT ON PRODUCT_DIM
  FOR EACH ROW
BEGIN
  SELECT PRODUCT_DIM_S.NEXTVAL
  INTO :NEW.PRODUCT_DIM_ID
  FROM DUAL;
END;
```

1.3. Naredba za kreiranje dimenzije DATE_DIM

```
CREATE TABLE DATE_DIM
(
  DATE_DIM_ID NUMBER(*,0) PRIMARY KEY,
```

```
        SALES_DATE DATE,
        SALES_DAY_MONTH NUMBER(*,0),
        SALES_MONTH NUMBER(*,0),
        SALES_YEAR NUMBER(*,0),
        SALES_QUARTER NUMBER(*,0)
    );

CREATE SEQUENCE DATE_DIM_S;

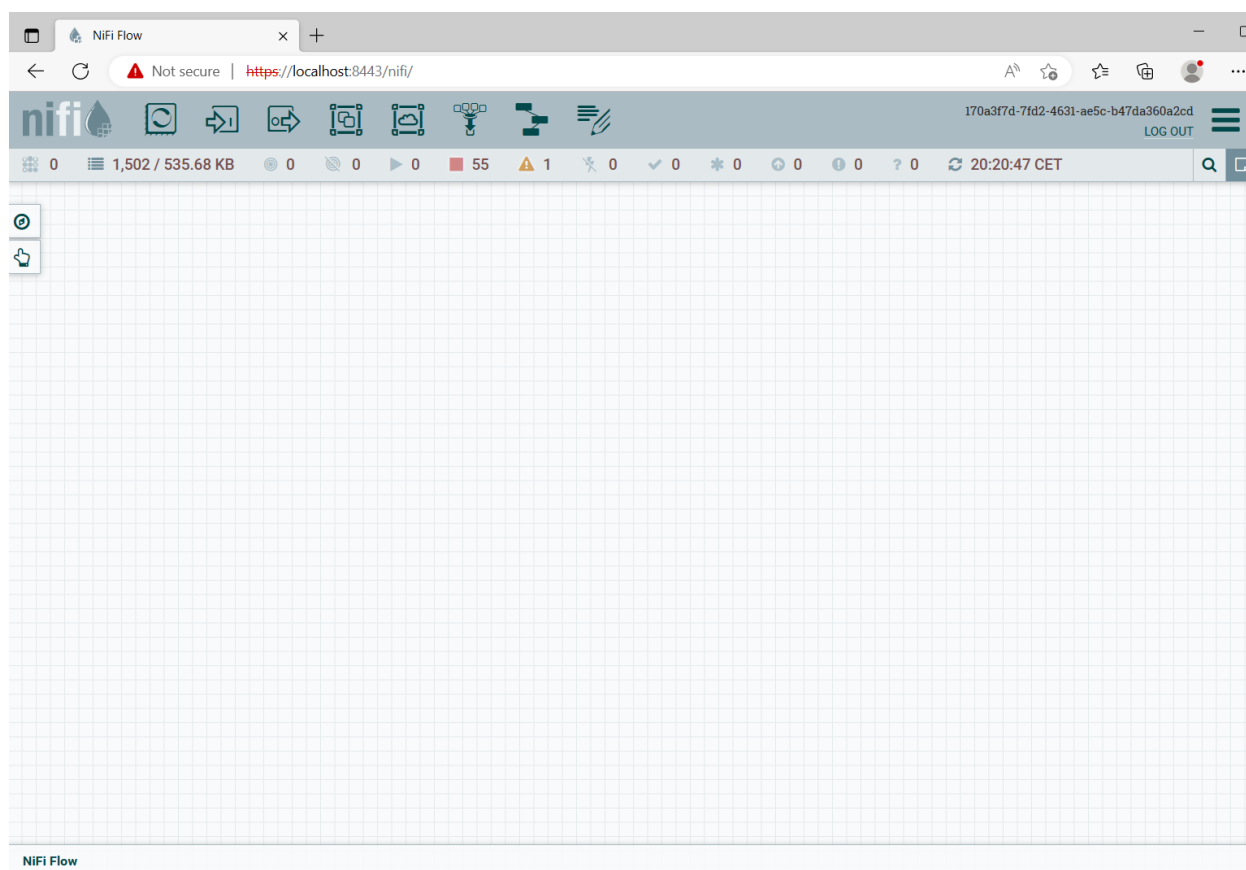
CREATE OR REPLACE TRIGGER date_dim_on_insert
    BEFORE INSERT ON DATE_DIM
    FOR EACH ROW
BEGIN
    SELECT DATE_DIM_S.NEXTVAL
    INTO :NEW.DATE_DIM_ID
    FROM DUAL;
END;
```

1.4. Naredba za kreiranje tabele činjenica SALES_FACT

```
CREATE TABLE SALES_FACT
(
    ORDER_ID NUMBER(*,0),
    CUSTOMER_DIM_ID NUMBER(*,0),
    PRODUCT_DIM_ID NUMBER(*,0),
    DATE_DIM_ID NUMBER(*,0),
    TOTAL_VALUE NUMBER,
    QUANTITY NUMBER,
    PRIMARY KEY (ORDER_ID, CUSTOMER_DIM_ID, PRODUCT_DIM_ID,
DATE_DIM_ID)
);
```

Pokretanje Apache NIFI

1. Pokrenuti Apache NIFI startovanjem **run-nifi.bat** programa koji se nalazi u **bin** folderu.
2. U Web pretraživaču otvoriti <https://localhost:8443/nifi/>
3. Uneti korisničko ime i lozinku koju je NIFI generisao. Nakon uspešne prijave otvoriće se prozor kao sa slike koja je data u nastavku.



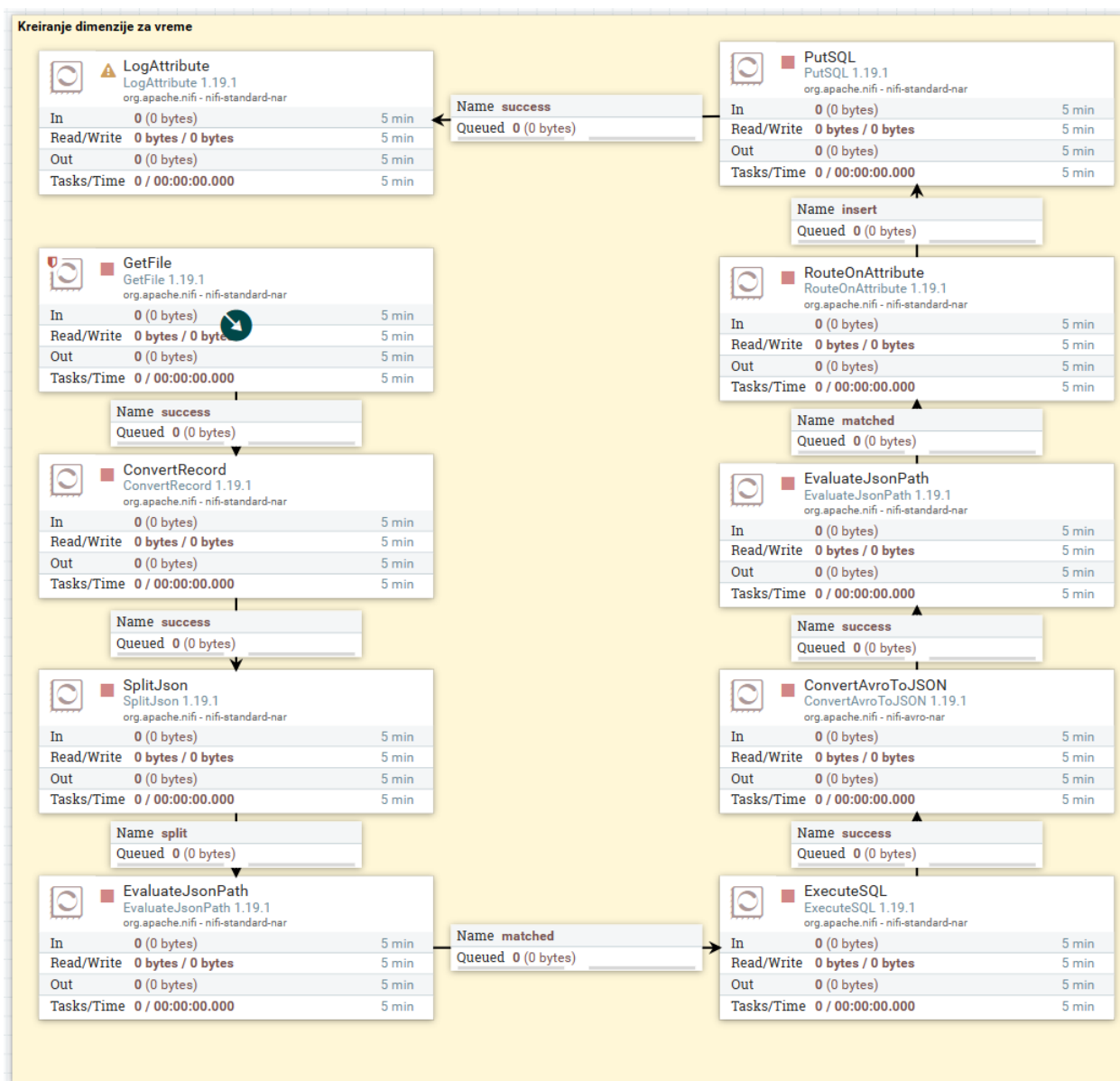
Kreiranje tokova podataka za popunjavanje dimenzionih tabela

Dimenzija za vreme

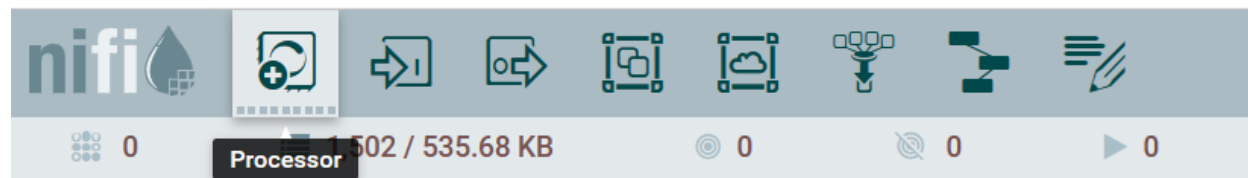
Podaci koje je potrebno upisati u dimenziju za vreme biće učitani iz CSV fajla koji je dat u okviru pripreme za ovu laboratorijsku vežbu. CSV fajl sadrži pet kolona:

- Date - koja sadrži datum
- Day in month - numerički podatak koji sadrži informaciju o tome koji je dan u mesecu
- Month - numerički podatak koji sadrži informaciju o tome koji je mesec u godini
- Year - sadrži godinu datuma
- Quartal - numerički podatak koji sadrži informaciju kom kvartalu datum pripada

Tok podataka koji je potrebno kreirati kako bi se izvršilo popunjavanje dimenzije za vreme dat je u nastavku.



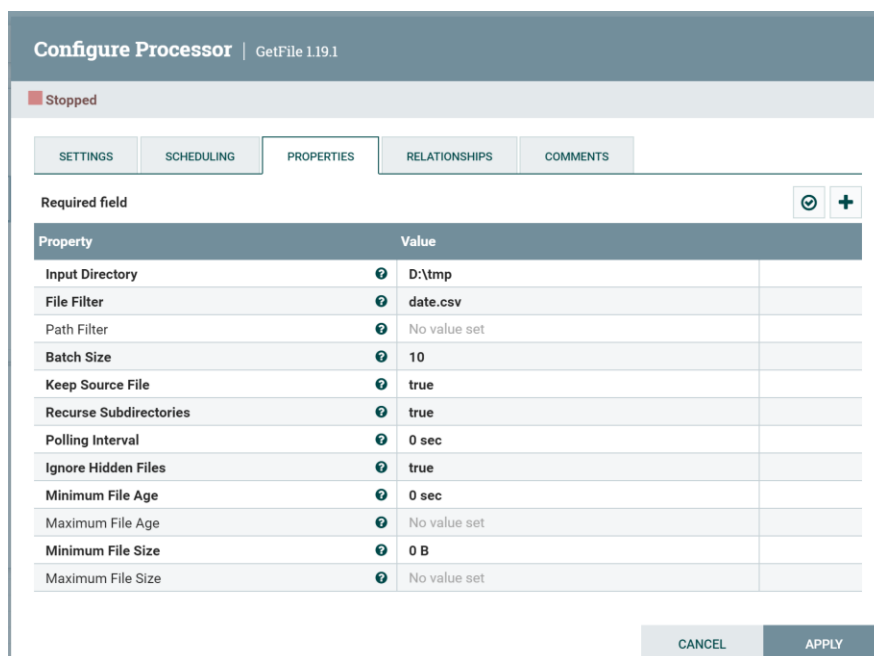
1. Prvi korak u kreiranju toka podataka za popunjavanje dimenzije za vreme je učitavanje podataka iz fajla.
 - 1.1. Prevući procesor na polje za kreiranje tokova podataka.



- 1.2. Nakon prevlačenja procesora otvara se prozor za izbor procesora koji treba da se doda. Izabrati procesor GetFile i kliknuti na ADD. Procesor GetFile kreira jedan FlowFile na izlazu koji sadrži sve podatke iz fajla.



- 1.3. Nakon dodavanja procesora, duplim klikom na procesor otvaraju se podešavanja.
- 1.4. U okviru podkartice PROPERTIES u polje Input Directory uneti putanju do foldera u kome se nalazi CSV fajl sa podacima a u polje File Filter upisati ime fajla.
- 1.5. Kliknuti na APPLY.



2. Podaci koji se nalaze u FlowFile su u CSV formatu. U sledećem karku konvertovaćemo ove podatke u JSON format.

2.1. Dodati procesor ConvertRecord.

2.2. Napraviti vezu od procesora GetFile do procesora ConvertRecord. Kada veza bude kreirana otvoriće se podešavanja veze. Podesiti For Relationships (podešavanje relacija veze) na success, čime se definiše da će FlowFile-ovi proći kroz ovu vezu u slučaju uspešnog izvršavanja GetFile. Kliknuti na ADD.

The screenshot shows the 'Create Connection' dialog box. It has a header bar with the title 'Create Connection'. Below the header, there are two tabs: 'DETAILS' and 'SETTINGS'. The 'DETAILS' tab is active. It contains a table with two columns: 'From Processor' and 'To Processor'. The 'From Processor' column has the value 'GetFile' and the 'To Processor' column has the value 'ConvertRecord'. Below the table, there is a section 'Within Group' with the value 'NiFi Flow'. At the bottom, there is a section 'For Relationships' with a checked checkbox and the value 'success'. At the bottom right, there are two buttons: 'CANCEL' and 'ADD'.

2.3. Otvoriti podešavanja ConvertRecord procesora.

2.4. Otvoriti karticu RELATIONSHIPS i odabrati da u slučaju neuspešnog ishoda konverzije FlowFile-a bude urađen prekid (opciju failure postavite na terminate)

2.5. Otvoriti karticu PROPERTIES

2.6. Podešavanje parametra Record Reader

2.6.1. Kliknuti na vrednost za podešavanje parametra

2.6.2. Iz padajućeg menija odabrati Create New Service

2.6.3. Otvoriće se prozor Add Controller Service

2.6.4. Iz padajućeg menija Compatible Controller Services odabrati CSVReader 1.19.1

2.6.5. Kliknuti na Create

2.6.6. Dobićete prozor kao u nastavku. Kliknuti na Go To strelicu pored Record Reader parametra.

Configure Processor | ConvertRecord 1.19.1

Invalid

SETTINGS SCHEDULING **PROPERTIES** RELATIONSHIPS COMMENTS

Required field

Property	Value
Record Reader	CSVReader
Record Writer	No value set
Include Zero Record FlowFiles	true

CANCEL APPLY

2.6.7. Klikom na Go To strelicu otvara se CONTROLLER SERVICES kartica. Dodati Record Reader označen je žutom bojom na slici.

NiFi Flow Configuration

GENERAL **CONTROLLER SERVICES**

Name	Type	Bundle	State	Scope
AvroRecordSetWriter	AvroRecordSetWriter 1.19.1	org.apache.nifi - nifi-recor...	Enabled	NiFi Flow
CSVReader	CSVReader 1.19.1	org.apache.nifi - nifi-recor...	Disabled	NiFi Flow
CSVReader	CSVReader 1.19.1	org.apache.nifi - nifi-recor...	Enabled	NiFi Flow
JsonRecordSetWriter	JsonRecordSetWriter 1.19...	org.apache.nifi - nifi-recor...	Enabled	NiFi Flow
Oracle	DBCPConnectionPool 1.1...	org.apache.nifi - nifi-dbc...	Enabled	NiFi Flow

Last updated: 11:53:33 CET

Listed services are available to all descendant Processors and services of this Process Group.

2.6.8. Otvoriti podešavanja i dodati format za datum koji odgovara formatu datuma iz CSV fajla (yyyy-MM-dd). Kliknuti na OK.

2.6.9. Promeniti status servisa na Enabled.

Controller Service Details
CSVReader 1.19.1

SETTINGS
PROPERTIES
COMMENTS

Required field

Property	Value
Schema Access Strategy	Infer Schema
CSV Parser	Apache Commons CSV
Date Format	yyyy-MM-dd
Time Format	No value set
Timestamp Format	No value set
CSV Format	Custom Format
Value Separator	,
Record Separator	\n
Treat First Line as Header	false
Ignore CSV Header Column Names	true
Quote Character	"
Escape Character	\
Comment Marker	No value set
Null String	No value set

OK

2.7. Podešavanje parametra Record Writer

2.7.1. Kliknuti na vrednost za podešavanje parametra.

2.7.2. Iz padajućeg menija odabrati Create New Service.

2.7.3. Otvoriće se prozor Add Controller Service.

2.7.4. Iz padajućeg menija Compatible Controller Services odabrati JsonRecordSetWriter.

2.7.5. Kliknuti na Create.

2.7.6. Otvoriti listu svih servisa klikom na strelicu Go To u nastavku vrednosti parametra.

2.7.7. Otvoriti podešavanja JsonRecordSetWriter servisa

2.7.8. Podesiti Date i Timestamp format

Controller Service Details
JsonRecordSetWriter 1.19.1

SETTINGS
PROPERTIES
COMMENTS

Required field

Property	Value
Schema Write Strategy	Do Not Write Schema
Schema Cache	No value set
Schema Access Strategy	Inherit Record Schema
Date Format	yyyy-MM-dd
Time Format	No value set
Timestamp Format	yyyy-MM-dd
Pretty Print JSON	false
Suppress Null Values	Never Suppress
Output Grouping	Array
Compression Format	none

OK

2.7.9. Promeniti status servisa na Enable.

3. Nakon konvertovanja formata podataka iz Flow File-a, na izlazu se dobija Flow File koji sadrži niz JSON objekata.
4. U sledećem koraku ovaj niz JSON objekata podelićemo tako da svaki objekat bude poseban Flow File.
 - 4.1. Dodati SplitJson procesor.
 - 4.2. Procesor ConvertRecord povezati sa SplitJson procesorom i odabrati success vezu.
 - 4.3. Otvoriti podešavanja SplitJson procesora.
 - 4.4. Otvoriti karticu PROPERTIES.
 - 4.5. Podešavanju JsonPath Expression postaviti vrednost \$.*
 - 4.6. Otvoriti karticu RELATIONSHIPS i odabrati opciju terminate za failure i original veze.
 - 4.7. Kliknuti na APPLY.

Configure Processor | Split.Json 1.19.1

Stopped

SETTINGS SCHEDULING **PROPERTIES** RELATIONSHIPS COMMENTS

Required field

Property	Value
JsonPath Expression	\$.*
Null Value Representation	empty string

CANCEL APPLY

5. U narednom koraku dodaju se novi FlowFile atributi koji pamte informacije iz JSON objekta.
 - 5.1. Dodati EvaluateJsonPath procesor.
 - 5.2. Procesor SplitJson povezati sa EvaluateJsonPath procesorom i odabrati spit vezu.
 - 5.3. Otvoriti podešavanja EvaluateJsonPath procesora.
 - 5.4. Otvoriti karticu PROPERTIES.
 - 5.5. Parametru Destination dodeliti vrednost flowfile-attribute čime se definiše da izlaz JsonPath evaluacije bude upisan kao atribut FlowFile-a.
 - 5.6. Klikom na znak plus dodaju se novi atributi. Dodati attribute SALES_DATE, SALES_DAY_MONTH, SALES_MONTH, SALES_QUARTER, SALES_YEAR. Kao vrednosti ovih atributa upisuju se vrednosti iz JSON objekta.

Configure Processor | EvaluateJsonPath 1.19.1

Stopped

SETTINGS SCHEDULING **PROPERTIES** RELATIONSHIPS COMMENTS

Required field 🔍 +

Property	Value	
Destination	flowfile-attribute	
Return Type	auto-detect	
Path Not Found Behavior	ignore	
Null Value Representation	empty string	
SALES_DATE	\$.SALES_DATE	🗑️
SALES_DAY_MONTH	\$.SALES_DAY_MONTH	🗑️
SALES_MONTH	\$.SALES_MONTH	🗑️
SALES_QUARTER	\$.SALES_QUARTER	🗑️
SALES_YEAR	\$.SALES_YEAR	🗑️

CANCEL APPLY

5.7. U okviru kartice RELATIONSHIPS i odabrati opciju terminate za failure i unmatched veze.

5.8. Kliknuti na APPLY.

Configure Processor | EvaluateJsonPath 1.19.1

Stopped

SETTINGS SCHEDULING PROPERTIES **RELATIONSHIPS** COMMENTS

Automatically Terminate / Retry Relationships 🔍

failure

☒ terminate ☐ retry

FlowFiles are routed to this relationship when the JsonPath cannot be evaluated against the content of the FlowFile; for instance, if the FlowFile is not valid JSON

matched

☐ terminate ☐ retry

FlowFiles are routed to this relationship when the JsonPath is successfully evaluated and the FlowFile is modified as a result

unmatched

☒ terminate ☐ retry

FlowFiles are routed to this relationship when the JsonPath does not match the content of the FlowFile and the Destination is set to flowfile-content

CANCEL APPLY

6. U dimenziju za vreme potrebno je dodavati samo nove datume i nema ažuriranja već upisanih vrednosti. Iz tog razloga, potrebno je izvršiti proveru za svaki od datuma u FlowFile-ovima da li se nalaze u dimenziji za vreme. Proveru da li je datum unet u dimenziju izvršićemo korišćenjem ExecuteSQL procesora. Ovaj procesor se koristi za izvršavanje SELECT SQL upita koji se definišu unutar procesora. Rezultat SQL upita vraća se u Avro formatu.

6.1. Dodati ExecuteSQL procesor.

6.2. Procesor EvaluateJsonPath povezati sa ExecuteSQL procesorom i odabrati matched vezu.

- 6.3. Otvoriti podešavanja ExecuteSQL procesora.
- 6.4. U okviru kartice RELATIONSHIPS i odabrati opciju terminate za failure vezu.
- 6.5. U okviru kartice PROPERTIES izvršiti podešavanja konekcije ka bazi i SQL naredbe koju je potrebno izvršiti.
 - 6.5.1. Database Connection Pooling Service se koristi za definisanje konekcije ka bazi. Kliknuti na vrednost ovog parametra i odabrati Create new service...
 - 6.5.2. U okviru prozora Add Controller Service odabrati DBCPConnectionPool 1.19.1. Kliknuti na CREATE.

Add Controller Service

Requires Controller Service
DBCPService 1.19.1 from org.apache.nifi - nifi-standard-services-api-nar

Compatible Controller Services
DBCPConnectionPool 1.19.1

Controller Service Name
DBCPConnectionPool

Bundle
org.apache.nifi - nifi-dbcp-service-nar

Tags
database, pooling, dbcp, jdbc, connection, store

Description

CANCEL CREATE

- 6.5.3. Otvoriti podešavanja DBCPConnectionPool-a klikom na GoTo strelicu.
- 6.5.4. Otvoriti podešavanja servisa.
- 6.5.5. U okviru kartice SETTINGS može da se promeni ime podešavanjem vrednosti za Name.

Configure Controller Service | DBCPConnectionPool 1.19.1

SETTINGS PROPERTIES COMMENTS

Name
DBCPConnectionPool

Id
f21f55cc-0186-1000-8fae-5b290f3aa689

Type
DBCPConnectionPool 1.19.1

Bundle
org.apache.nifi - nifi-dbcp-service-nar

Supports Controller Service
• DBCPService 1.19.1 from org.apache.nifi - nifi-standard-services-api-nar

Bulletin Level
WARN

Referencing Components
▼ Processors (1)
▲ ExecuteSQL ExecuteSQL

CANCEL APPLY

- 6.5.6. U okviru kartice PROPERTIES vrši se podešavanje konekcije.

6.5.7. U polje Database Connection URL upisati parameter za pristup bazi podataka:

jdbc:oracle:thin:dwuser/ipzuser@160.99.12.92:1521/GISLAB_PD

6.5.8. U polje Database Driver Class Name upisati: **oracle.jdbc.OracleDriver**

6.5.9. U polje Database Driver Location(s) uneti lokaciju na kojoj se drajver nalazi:

lib/ojdbc8.jar

Controller Service Details | DBCPConnectionPool 1.19.1

SETTINGS | **PROPERTIES** | COMMENTS

Required field

Property	Value
Database Connection URL	jdbc:oracle:thin:dwuser/dwuser23@160.99.12.92:1521...
Database Driver Class Name	oracle.jdbc.OracleDriver
Database Driver Location(s)	lib/ojdbc8.jar
Kerberos User Service	No value set
Kerberos Credentials Service	No value set
Kerberos Principal	No value set
Kerberos Password	No value set
Database User	No value set
Password	Sensitive value set
Max Wait Time	500 millis
Max Total Connections	8
Validation query	No value set
Minimum Idle Connections	0
Max Idle Connections	8

OK

6.5.10. Kliknuti na OK.

6.5.11. Promeniti status servisa na Enabled. Zatvoriti prozor.

6.5.12. U okviru kartice PROPERTIES ExecuteSQL procesora uneti SQL naredbu koja treba da se izvrši u polje SQL select query:

```
SELECT DATE_DIM_ID AS PK_DIM
FROM DATE_DIM
WHERE SALES_DATE = to_date('${SALES_DATE}', 'yyyy-MM-dd')
```

Upit vraća vrednosti primarnog ključa ukoliko traženi datum postoji u bazi podataka kao sadržaj FlowFile-a.

6.5.13. Kliknuti na APPLY i zatvoriti podešavanja procesora.

7. Izlaz ovog procesora iz Avro formata prebacujemo u JSON format korišćenjem ConvertAvroToJson procesora.

7.1. Dodati ConvertAvroToJson procesor.

7.2. Procesor ExecuteSQL povezati sa ConvertAvroToJson procesorom i odabrati success vezu.

7.3. Otvoriti podešavanja ExecuteSQL procesora.

7.4. U okviru kartice RELATIONSHIPS, odabrati opciju terminate za failure vezu.

7.5. Kliknuti na APPLY.

8. Koristimo EvaluateJsonPath procesor za dodavanje atributa PK_DIM FlowFile-u.

- 8.1. Dodati EvaluateJsonPath procesor.
- 8.2. Procesor ConvertAvroToJson povezati sa EvaluateJsonPath procesorom i odabrati success vezu.
- 8.3. Otvoriti podešavanja EvaluateJsonPath procesora.
- 8.4. Otvoriti karticu PROPERTIES.
- 8.5. Parametru Destination dodeliti vrednost flowfile-attribute čime se definiše da izlaz JsonPath evaluacije bude upisan kao atribut FlowFile-a.
- 8.6. Dodati atribut PK_DIM.
- 8.7. U okviru kartice RELATIONSHIPS, odabrati opciju terminate za failure i unmatched veze.
- 8.8. Kliknuti na APPLY.

Configure Processor | EvaluateJsonPath 1.19.1

Invalid

SETTINGS | SCHEDULING | PROPERTIES | RELATIONSHIPS | COMMENTS

Required field

Property	Value
Destination	flowfile-attribute
Return Type	auto-detect
Path Not Found Behavior	ignore
Null Value Representation	empty string
PK_DIM	\$.PK_DIM

CANCEL | APPLY

9. U sledećem koraku radimo rutiranje podataka. Cilj ovog koraka je da proveriti da li datum postoji u bazi i ukoliko ne postoji da ga prosledi dalje procesoru koju će izvršiti upis datuma u tabelu. Ova provera radi se na osnovu vrednosti atributa Flow File-a PK_DIM. Ukoliko je atribut prazan, podatak ne postoji u bazi podataka i treba ga proslediti dalje. Ukoliko atribut ima vrednost, datum je već upisan i ne treba da bude prosleđen dalje procesoru za upis u bazu podataka. U ovom koraku koristimo RouteOnAttribute procesor.
- 9.1. Dodati RouteOnAttribute procesor.
- 9.2. Procesor EvaluateJsonPath povezati sa RouteOnAttribute procesorom i odabrati matched vezu.
- 9.3. Otvoriti podešavanja RouteOnAttribute procesora.
- 9.4. Otvoriti karticu PROPERTIES.
- 9.5. Klikom na Add Property dodati uslov za rutiranje podataka:
Property Name: insert
Vrednost: \${ PK_DIM>equals("")}
- 9.6. U okviru kartice RELATIONSHIPS, odabrati opciju terminate za unmatched vezu.

9.7. Kliknuti na APPLY.

Configure Processor | RouteOnAttribute 1.19.1

Stopped

SETTINGS SCHEDULING **PROPERTIES** RELATIONSHIPS COMMENTS

Required field

Property	Value
Routing Strategy	Route to Property name
insert	<code>\${PK_DIM:equals(')}</code>

CANCEL APPLY

10. U posljednjem koraku radi se upisivanje podataka u tabelu dimenzije za vreme. Za upisivanje podataka koristi se PutSQL procesor. Ovaj procesor izvršava SQL UPDATE ili INSERT naredbe. SQL naredba koju treba procesor da izvrši može da se definiše u podešavanjima procesora ili da bude prosleđena kao sadržaj FlowFile-a. U ovom primeru SQL naredba biće definisana u procesoru.

10.1. Dodati PutSQL procesor.

10.2. Otvoriti podešavanja procesora.

10.3. Otvoriti karticu PROPERTIES.

10.4. Kreirati novu konekciju ka Oracle bazi podataka koja ukazuje na lokaciju Data Mart-a kao u koraku 6.5.

10.5. Postaviti Support Fragmented Transactions na false.

10.6. U polje SQL Statement uneti SQL naredbu:

```
insert into DATE_DIM (SALES_DATE, SALES_DAY_MONTH, SALES_MONTH,
SALES_YEAR, SALES_QUARTER) VALUES (to_date('${SALES_DATE}',
'yyyy-MM-dd'), ${SALES_DAY_MONTH}, ${SALES_MONTH}, ${SALES_YEAR},
${SALES_QUARTER})
```

10.7. Kliknuti na APPLY.

Configure Processor | PutSQL 1.19.1

Stopped

SETTINGS | SCHEDULING | **PROPERTIES** | RELATIONSHIPS | COMMENTS

Required field ⓘ +

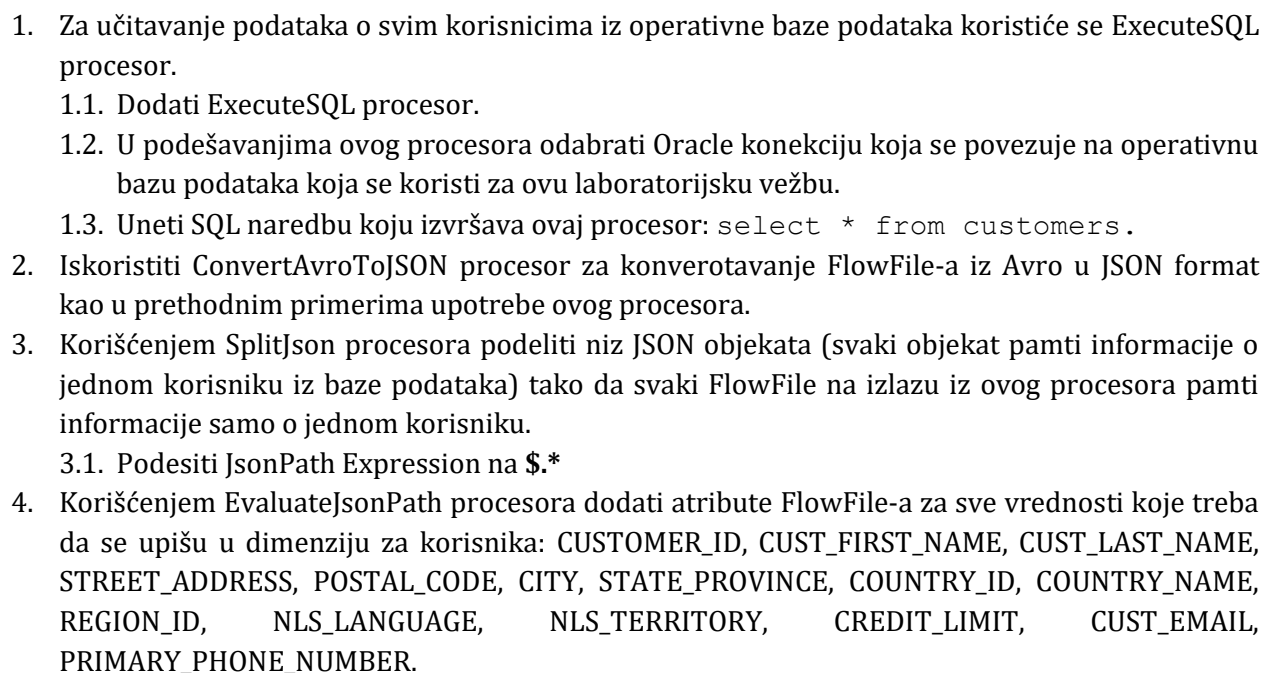
Property	Value
JDBC Connection Pool	Oracle →
SQL Statement	insert into DATE_DIM_2 (SALES_DATE, SALES_D...
Support Fragmented Transactions	false
Database Session AutoCommit	false
Transaction Timeout	No value set
Batch Size	1
Obtain Generated Keys	false
Rollback On Failure	false

CANCEL APPLY

Dimenzija za pamćenje informacija o korisnicima

Podaci koje je potrebno upisati u dimenziju za pamćenje informacija o korisnicima učitavaju se iz Oracle baze podataka, iz tabele CUSTOMERS. Za potrebe ove vežbe, u tabeli dimenzije će se pamti sledeće informacije o korisnicima: CUSTOMER_ID, CUST_FIRST_NAME, CUST_LAST_NAME, STREET_ADDRESS, POSTAL_CODE, CITY, STATE_PROVINCE, COUNTRY_ID, COUNTRY_NAME, REGION_ID, NLS_LANGUAGE, NLS_TERRITORY, CREDIT_LIMIT, CUST_EMAIL, PRIMARY_PHONE_NUMBER. Koristiće se Type 2 Slowly Changing Dimension za rešavanje problema promena dimenzija.

Tok podataka koji popunjava dimenziju za pamćenje informacija o korisnicima dat je u nastavku.



Configure Processor | EvaluateJsonPath 1.19.1

Stopped

SETTINGS SCHEDULING **PROPERTIES** RELATIONSHIPS COMMENTS

Required field

Property	Value
Destination	flowfile-attribute
Return Type	auto-detect
Path Not Found Behavior	ignore
Null Value Representation	empty string
CITY	\$.CITY
COUNTRY_ID	\$.COUNTRY_ID
COUNTRY_NAME	\$.COUNTRY_NAME
CREDIT_LIMIT	\$.CREDIT_LIMIT
CUST_EMAIL	\$.CUST_EMAIL
CUST_FIRST_NAME	\$.CUST_FIRST_NAME
CUST_LAST_NAME	\$.CUST_LAST_NAME
CUSTOMER_ID	\$.CUSTOMER_ID

CANCEL APPLY

5. Korišćenjem ExecuteSQL procesora za svaki FlowFile proveriti da li se korisnik već nalazi u tabeli dimenzije:

```
SELECT *
FROM CUSTOMER_DIM
WHERE CUSTOMER_ID = ${CUSTOMER_ID} AND DATE_TO IS NULL
```

6. Izlaz ovog procesora konvertovati iz Avro formata u JSON korišćenjem ConvertAvroToJson procesora.
7. Korišćenjem EvaluateJsonPath procesora dodati atribut FlowFile-a za sve kolone koje je vratila SELECT naredba izvršena u koraku 5 a koje se upisuju u dimenziju za korisnike.

```
CITY_R : $.CITY
COUNTRY_NAME_R: $.COUNTRY_NAME
CREDIT_LIMIT_R : $.CREDIT_LIMIT
CUST_EMAIL_R : $.CUST_EMAIL
CUST_FIRST_NAME_R :$.CUST_FIRST_NAME
CUST_LAST_NAME_R :$.CUST_LAST_NAME
CUSTOMER_DIM_ID : $.CUSTOMER_DIM_ID
CUSTOMER_ID_R : $.CUSTOMER_ID
DATE_FROM : $.DATE_FROM
DATE_TO : $.DATE_TO
NLS_LANGUAGE_R : $.NLS_LANGUAGE
NLS_TERRITORY_R : $.NLS_TERRITORY
POSTAL_CODE_R : $.POSTAL_CODE
PRIMARY_PHONE_NUMBER_R : $.PRIMARY_PHONE_NUMBER
REGION_ID_R : $.REGION_ID
STATE_PROVINCE_R : $.STATE_PROVINCE
STREET_ADDRESS_R : $.STREET_ADDRESS
```

8. U ovom koraku može da se pristupi rutiranju podataka. Na osnovu vrednosti koje su upisane u attribute FlowFile-a može da se proverí šta treba uraditi sa podatkom:

- Upisati ga u dimenziju ukoliko ne postoji u tabeli CUSTOMER_DIM
- Izvršiti ažuriranje korišćenjem Type 2 Slowly Changing Dimension pristupa za ažuriranje podataka
- Ne raditi ništa ukoliko se podatak nalazi u dimenziji i ni jedna kolona nema izmenjenu vrednosti

8.1. Dodati RouteOnAttribute procesor

8.2. U okviru kartice PROPERTIES klikom na znak "+" (Add property) dodati dva nova svojstva:

8.2.1. Proverava da li je select naredba izvršena u koraku 5 vratila torku iz tabele, odnosno da li je korisnik već upisan u dimenziju

```
insert : ${CUSTOMER_DIM_ID:isEmpty() }
```

8.2.2. Provera da li neka od kolona ima promenjenu vrednost, odnosno da li podatak treba da se ažurira u dimenziji

```
no_change: ${CUSTOMER_ID>equals(${CUSTOMER_ID_R}):and(
  ${CUST_FIRST_NAME>equals(${CUST_FIRST_NAME_R}):and(
    ${CUST_LAST_NAME>equals(${CUST_LAST_NAME_R}):and(
      ${STREET_ADDRESS>equals(${STREET_ADDRESS_R}):and(
        ${POSTAL_CODE>equals(${POSTAL_CODE_R}):and(
          ${CITY>equals(${CITY_R}):and(
            ${STATE_PROVINCE>equals(${STATE_PROVINCE_R}):and(
              ${COUNTRY_ID>equals(${COUNTRY_ID_R}):and(
                ${COUNTRY_NAME>equals(${COUNTRY_NAME_R}):and(
                  ${REGION_ID>equals(${REGION_ID_R}):and(
                    ${NLS_LANGUAGE>equals(${NLS_LANGUAGE_R}):and(
                      ${NLS_TERRITORY>equals(${NLS_TERRITORY_R}):and(
                        ${CREDIT_LIMIT>equals(${CREDIT_LIMIT_R}):and(
                          ${CUST_EMAIL>equals(${CUST_EMAIL_R}):and(
                            ${PRIMARY_PHONE_NUMBER>equals(${PRIMARY_PHONE_NUMBER_R}):and(
                              d(${CUSTOMER_DIM_ID:isEmpty():not()})
                            )
                          )
                        )
                      )
                    )
                  )
                )
              )
            )
          )
        )
      )
    )
  )
}
```

8.3. Kreirana svojstva biće iskorišćena za rutiranje podataka. Kroz insert vezu proćiće podaci koje je potrebno upisati u bazu, no_change vezu podaci kojima nije potrebna dalja obrada i unmatched vezu (relacija koja se koristi ukoliko FlowFile ne odgovara ni jednom od definisanih izraza) podaci koje je potrebno ažurirati.

Configure Processor | RouteOnAttribute 1.19.1

Stopped

SETTINGS SCHEDULING **PROPERTIES** RELATIONSHIPS COMMENTS

Required field

Property	Value
Routing Strategy	Route to Property name
insert	\${CUSTOMER_DIM_ID.isEmpty()}
no_change	\${CUSTOMER_ID.equals(CUSTOMER_ID_R)}:and(...)

CANCEL APPLY

9. Dodavanje novog korisnika u dimenziju.

9.1. Dodati PutSQL procesor.

9.2. Procesor RouteOnAttribute povezati sa PutSQL procesorom i odabrati insert vezu.

9.3. Otvoriti podešavanja procesora – karticu PROPERTIES

9.4. Odabrati konekciju ka Oracle bazi podataka u kojoj se nalazi Data Mart.

9.5. Postaviti Support Fragmented Transactions na false.

9.6. U polje SQL Statement uneti naredbu za unos novog korisnika u dimenziju:

```
INSERT INTO CUSTOMER_DIM (DATE_FROM, DATE_TO, CUSTOMER_ID,
CUST_FIRST_NAME, CUST_LAST_NAME, STREET_ADDRESS, POSTAL_CODE, CITY,
STATE_PROVINCE, COUNTRY_ID, COUNTRY_NAME, REGION_ID, NLS_LANGUAGE,
NLS_TERRITORY, CREDIT_LIMIT, CUST_EMAIL, PRIMARY_PHONE_NUMBER)
VALUES ('01-JAN-1900', NULL, ${CUSTOMER_ID}, '${CUST_FIRST_NAME}',
'${CUST_LAST_NAME}', '${STREET_ADDRESS}', '${POSTAL_CODE}', '${CITY}',
'${STATE_PROVINCE}', '${COUNTRY_ID}', '${COUNTRY_NAME}',
'${REGION_ID}', '${NLS_LANGUAGE}', '${NLS_TERRITORY}',
'${CREDIT_LIMIT}', '${CUST_EMAIL}', '${PRIMARY_PHONE_NUMBER}')
```

10. Ažuriranje korisnika u dimenziji

10.1. Dodati PutSQL procesor.

10.2. Procesor RouteOnAttribute povezati sa PutSQL procesorom i odabrati unmatched vezu.

10.3. Otvoriti podešavanja procesora – karticu PROPERTIES.

10.4. Odabrati konekciju ka Oracle bazi podataka u kojoj se nalazi Data Mart.

10.5. Postaviti Support Fragmented Transactions na false.

10.6. U polje SQL Statement uneti naredbu koja ažurira postojeću aktivnu torku korisnika u dimenziji i postavlja vrednost kolone DATE_TO na tekući datum:

```
UPDATE CUSTOMER_DIM
SET DATE_TO = '${now():format('dd-MMM-yy')}'
WHERE CUSTOMER_DIM_ID = ${CUSTOMER_DIM_ID}
```

- 10.7. Dodati PutSQL procesor koji će biti korišćen za unošenje nove torke za korisnika sa ažuriranim informacijama.
- 10.8. Procesor PutSQL koji ažurira postojeću torku u dimenziji (korak 10.1) povezati sa PutSQL procesorom koji dodaje novu torku za korisnika (korak 10.7) i odabrati success vezu.
- 10.9. Odabrati konekciju ka Oracle bazi podataka u kojoj se nalazi Data Mart.
- 10.10. Postaviti Support Fragmented Transactions na false.
- 10.11. U polje SQL Statement uneti naredbu za dodavanje korisnika sa ažuriranim informacijama:

```
INSERT INTO CUSTOMER_DIM (DATE_FROM, DATE_TO, CUSTOMER_ID,  
CUST_FIRST_NAME, CUST_LAST_NAME, STREET_ADDRESS, POSTAL_CODE, CITY,  
STATE_PROVINCE, COUNTRY_ID, COUNTRY_NAME, REGION_ID, NLS_LANGUAGE,  
NLS_TERRITORY, CREDIT_LIMIT, CUST_EMAIL, PRIMARY_PHONE_NUMBER)  
VALUES ('${now():format('dd-MMM-yy')} ', NULL, ${CUSTOMER_ID},  
 '${CUST_FIRST_NAME}', '${CUST_LAST_NAME}', '${STREET_ADDRESS}',  
 '${POSTAL_CODE}', '${CITY}', '${STATE_PROVINCE}', '${COUNTRY_ID}',  
 '${COUNTRY_NAME}', '${REGION_ID}', '${NLS_LANGUAGE}',  
 '${NLS_TERRITORY}', '${CREDIT_LIMIT}', '${CUST_EMAIL}',  
 '${PRIMARY_PHONE_NUMBER}')
```

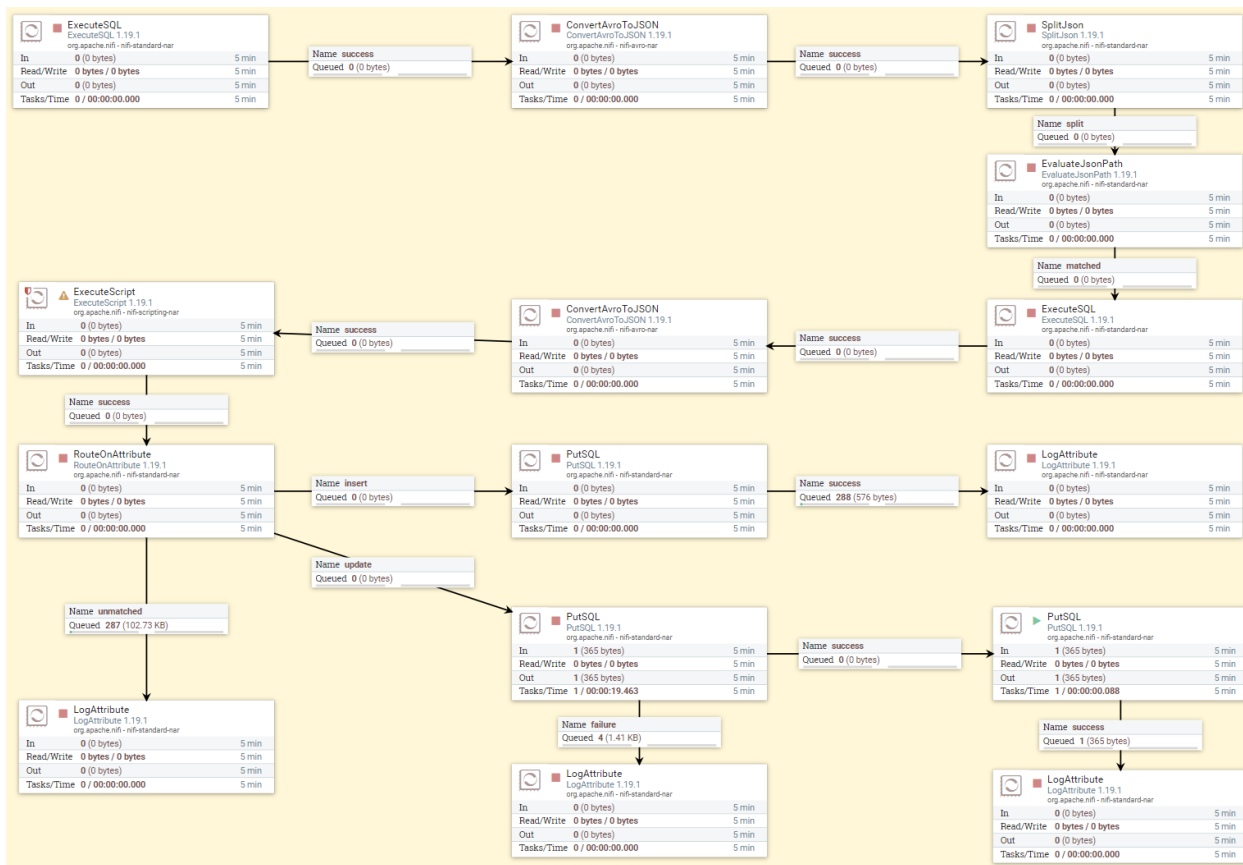
11. Dodavanje LogAttribute procesora

- 11.1. Dodati LogAttribute procesor.
- 11.2. Procesor RouteOnAttribute povezati sa LogAttribute procesorom i odabrati no_change vezu.

Dimenzija za pamćenje informacija o proizvodima

Podaci koje je potrebno upisati u dimenziju za pamćenje informacija o proizvodima učitavaju se iz Oracle baze podataka, iz tabele PRODUCTS. Za potrebe ove vežbe u tabeli dimenzije će se pamtit i sledeće informacije o korisnicima: PRODUCT_ID, PRODUCT_NAME, LANGUAGE_ID, MIN_PRICE, LIST_PRICE, SUPPLIER_ID, WARRANTY_PERIOD, WEIGHT_CLASS, CATEGORY_ID, PARENT_CATEGORY_ID i CATEGORY_NAME. Koristiće se Type 2 Slowly Changing Dimension za rešavanje problema promena dimenzija.

Tok podataka koji popunjava dimenziju za pamćenje informacija o korisnicima dat je u nastavku.



1. Za učitavanje podataka o svim proizvodima iz operativne baze podataka koristiće se ExecuteSQL procesor.
 - 1.1. Dodati ExecuteSQL procesor.
 - 1.2. U podešavanjima ovog procesora odabrati Oracle konekciju koja se povezuje na operativnu bazu podataka koja se koristi za ovu laboratorijsku vežbu.
 - 1.3. Uneti SQL naredbu koju izvršava ovaj procesor: `select * from products.`
2. Iskoristiti ConvertAvroToJSON procesor za konvertovanje FlowFile-a iz Avro u JSON format kao u prethodnim primerima upotrebe ovog procesora.
3. Korišćenjem SplitJson procesora podeliti niz JSON objekata (svaki objekat pamti informacije o jednom proizvodu iz baze podataka) tako da svaki FlowFile na izlazu iz ovog procesora pamti informacije samo o jednom proizvodu.
 - 3.1. Podesiti JsonPath Expression na `$.*`
4. Korišćenjem EvaluateJsonPath procesora dodati atribut FlowFile-a za sve vrednosti koje treba da se upišu u dimenziju za korisnika: PRODUCT_ID, PRODUCT_NAME, LANGUAGE_ID, MIN_PRICE, LIST_PRICE, SUPPLIER_ID, WARRANTY_PERIOD, WEIGHT_CLASS, CATEGORY_ID, PARENT_CATEGORY_ID i CATEGORY_NAME.

Configure Processor | EvaluateJsonPath 1.19.1

Stopped

SETTINGS SCHEDULING PROPERTIES RELATIONSHIPS COMMENTS

Required field

Property	Value
Destination	flowfile-attribute
Return Type	auto-detect
Path Not Found Behavior	ignore
Null Value Representation	empty string
CATEGORY_ID	\$.CATEGORY_ID
CATEGORY_NAME	\$.CATEGORY_NAME
LANGUAGE_ID	\$.LANGUAGE_ID
LIST_PRICE	\$.LIST_PRICE
MIN_PRICE	\$.MIN_PRICE
PARENT_CATEGORY_ID	\$.PARENT_CATEGORY_ID
PRODUCT_ID	\$.PRODUCT_ID
PRODUCT_NAME	\$.PRODUCT_NAME

CANCEL APPLY

5. Korišćenjem ExecuteSQL procesora za svaki FlowFile proveriti da li se proizvod već nalazi u tabeli dimenzije:

```
SELECT *
FROM PRODUCT_DIM
WHERE PRODUCT_ID = ${PRODUCT_ID} AND DATE_TO IS NULL
```

6. Izlaz ovog procesora konvertovati iz Avro formata u JSON korišćenjem ConvertAvroToJson procesora.
7. Dodati ExecuteScript procesor ([link](#)). Korišćenjem ovog procesora moguće je dodati kod koji je potrebno izvršiti. U okviru koda moguće je izvršiti akcije poput čitanja sadržaja i/ili atributa FlowFile-a, kreiranja novog FlowFile-a, unosa sadržaja i/ili atributa u FlowFile, interakcija sa ProcessSession-om kako bi se FlowFile prosledio odgovarajućoj vezi i dr.

U okviru ove vežbe ovaj procesor biće iskorišćen za izvršavanje Python koda sa ciljem da se izvrši provera da li proizvod već postoji u dimenziji i ukoliko postoji da li ga je potrebno ažurirati.

Procesor ima četiri svojstva:

Script Engine – u okviru koga se bira Script Engine koji se koristi.

Script File – kome se prosleđuje putanja do fajla u kome se nalazi kod koji je potrebno izvršiti. Moguće je definisati putanju do samo jednog fajla.

Script Body – unosi se kod koji je potrebno izvršiti.

Module Directory - lista putanja do datoteka i/ili direktorijuma razdvojenih zarezima koji sadrže module koje zahteva skripta.

Script File i Script Body ne mogu da se kombinuju u isto vreme.

- 7.1. Dodati ExecuteScript procesor.
- 7.2. Povezati ConvertAvroToJson sa ExecuteScript procesorom success vezom.
- 7.3. Otvoriti podešavanja ExecuteScript procesora.

7.4. U okviru kartice PROPERTIES odabrati Python kao Script Engine i uneti u polje Script Body kod koji se nalazi u nastavku:

```
from org.apache.commons.io import IOUtils
from java.nio.charset import StandardCharsets
from org.apache.nifi.processor.io import InputStreamCallback
import json
from java.lang import Object
from jarray import array

class PyInputStreamCallback(InputStreamCallback):
    def __init__(self):
        pass
    def process(self, inputStream):
        global loadedJsonObject
        text = IOUtils.toString(inputStream, StandardCharsets.UTF_8)
        loadedJsonObject = json.loads(text)
# end class

flowFile = session.get()
if (flowFile != None):

    PRODUCT_ID = flowFile.getAttribute('PRODUCT_ID')
    PRODUCT_NAME = flowFile.getAttribute('PRODUCT_NAME')
    CATEGORY_ID = flowFile.getAttribute('CATEGORY_ID')
    CATEGORY_NAME = flowFile.getAttribute('CATEGORY_NAME')
    LANGUAGE_ID = flowFile.getAttribute('LANGUAGE_ID')
    LIST_PRICE = flowFile.getAttribute('LIST_PRICE')
    MIN_PRICE = flowFile.getAttribute('MIN_PRICE')
    PARENT_CATEGORY_ID = flowFile.getAttribute('PARENT_CATEGORY_ID')
    SUPPLIER_ID = flowFile.getAttribute('SUPPLIER_ID')
    WARRANTY_PERIOD = flowFile.getAttribute('WARRANTY_PERIOD')
    WEIGHT_CLASS = flowFile.getAttribute('WEIGHT_CLASS')

    session.read(flowFile, PyInputStreamCallback())
    flowfile_json = loadedJsonObject

    product_exists = 'no'
    for_update = 'yes'
    PRODUCT_DIM_ID = ''
    if "PRODUCT_DIM_ID" in flowfile_json:
        product_exists = 'yes'

    PRODUCT_DIM_ID = flowfile_json['PRODUCT_DIM_ID']
    DIM_PRODUCT_ID = flowfile_json['PRODUCT_ID']
```

```

DIM_PRODUCT_NAME = flowfile_json['PRODUCT_NAME']
DIM_CATEGORY_ID = flowfile_json['CATEGORY_ID']
DIM_CATEGORY_NAME = flowfile_json['CATEGORY_NAME']
DIM_LANGUAGE_ID = flowfile_json['LANGUAGE_ID']
DIM_LIST_PRICE = flowfile_json['LIST_PRICE']
DIM_MIN_PRICE = flowfile_json['MIN_PRICE']
DIM_PARENT_CATEGORY_ID = flowfile_json['PARENT_CATEGORY_ID']
DIM_SUPPLIER_ID = flowfile_json['SUPPLIER_ID']
DIM_WARRANTY_PERIOD = flowfile_json['WARRANTY_PERIOD']
DIM_WEIGHT_CLASS = flowfile_json['WEIGHT_CLASS']

if (PRODUCT_ID==DIM_PRODUCT_ID and PRODUCT_NAME==DIM_PRODUCT_NAME and
CATEGORY_ID==DIM_CATEGORY_ID and CATEGORY_NAME==DIM_CATEGORY_NAME
    and LANGUAGE_ID==DIM_LANGUAGE_ID and LIST_PRICE==DIM_LIST_PRICE and
MIN_PRICE==DIM_MIN_PRICE and PARENT_CATEGORY_ID==DIM_PARENT_CATEGORY_ID
    and SUPPLIER_ID==DIM_SUPPLIER_ID and WARRANTY_PERIOD==DIM_WARRANTY_PERIOD and
WEIGHT_CLASS==DIM_WEIGHT_CLASS):
    for_update = 'no'

attrMap = {'product_exists':product_exists, 'for_update':for_update, 'PRODUCT_DIM_ID':
PRODUCT_DIM_ID}
flowFile = session.putAllAttributes(flowFile, attrMap)
session.transfer(flowFile, REL_SUCCESS)

```

Configure Processor
ExecuteScript 1.19.1

Stopped

SETTINGS
SCHEDULING
PROPERTIES
RELATIONSHIPS
COMMENTS

Required field

Property	Value
Script Engine	python
Script File	No value set
Script Body	from org.apache.commons.io import IOUtils...
Module Directory	No value set

CANCEL
APPLY

8. Kao rezultat izvršavanja Python koda korišćenjem ExecuteScript procesora svakom FlowFile-u su dodata 3 nova atributa koga će biti upotrebljena za rutiranje podataka:

- PRODUCT_DIM_ID - čuva primarni ključ dimenzije,
- product_exists – ima vrednost yes ili no u zavisnosti da li se proizvod već nalazi u dimenziji,
- for_update - ima vrednost yes ili no u zavisnosti da li proizvod treba ažurirati u tabeli dimenzije.

Za rutiranje FlowFile-ova koristiće se atributi product_exists i for_update.

8.1. Dodati RouteOnAttribute procesor

8.2. Povezati ExecuteSQL sa RouteOnAttribute procesorom success vezom.

8.3. Otvoriti podešavanja RouteOnAttribute procesora

8.4. U okviru kartice PROPERTIES klikom na znak "+" (Add property) dodati dva nova svojstva:
 8.4.1. Insert – proverava da li atribut product_exists ima vrednost "no", ukoliko ima vrednost "no", proizvod ne postoji u dimenziji i treba ga dodati

```
insert: ${product_exists>equals('no')}
```

8.4.2. Update – proverava da li atributi product_exists i for_update imaju vrednost "yes", ukoliko oba atributa imaju vrednost yes proizvod postoji u tabeli dimenzije ali ga je potrebno ažurirati.

```
update: ${product_exists>equals('yes'):and(  
  ${for_update>equals('yes')} ) }
```

8.5. Kreirana svojstva biće iskorišćena za rutiranje podataka. Kroz insert vezu proćiće podaci koje je potrebno upisati u bazu, update vezu podaci podaci koje je potrebno ažurirati, a unmatched vezu podaci kojima nije potrebna dalja obrada.

Configure Processor | RouteOnAttribute 1.19.1

Stopped

SETTINGS SCHEDULING PROPERTIES RELATIONSHIPS COMMENTS

Required field

Property	Value
Routing Strategy	Route to Property name
insert	<code>\${product_exists>equals('no')}</code>
update	<code>\${product_exists>equals('yes'):and(...</code>

CANCEL APPLY

9. Dodavanje proizvoda u dimenziju.

9.1. Dodati PutSQL procesor.

9.2. Procesor RouteOnAttribute povezati sa PutSQL procesorom i odabrati insert vezu.

9.3. Otvoriti podešavanja procesora – karticu PROPERTIES

9.4. Odabrati konekciju ka Oracle bazi podataka u kojoj se nalazi Data Mart.

9.5. Postaviti Support Fragmented Transactions na false.

9.6. U polje SQL Statement uneti naredbu za unos novog korisnika u dimenziju:

```
INSERT INTO PRODUCT_DIM (DATE_FROM, DATE_TO, PRODUCT_ID, PRODUCT_NAME,  
LANGUAGE_ID, MIN_PRICE, LIST_PRICE, SUPPLIER_ID, WARRANTY_PERIOD,  
WEIGHT_CLASS, CATEGORY_ID, PARENT_CATEGORY_ID, CATEGORY_NAME)  
VALUES ('01-JAN-1900', NULL, ${PRODUCT_ID}, '${PRODUCT_NAME}',  
 '${LANGUAGE_ID}', '${MIN_PRICE}', '${LIST_PRICE}', '${SUPPLIER_ID}',  
 '${WARRANTY_PERIOD}', '${WEIGHT_CLASS}', '${CATEGORY_ID}',  
 '${PARENT_CATEGORY_ID}', '${CATEGORY_NAME}')
```

10. Ažuriranje proizvoda u dimenziji

10.1. Dodati PutSQL procesor.

10.2. Procesor RouteOnAttribute povezati sa PutSQL procesorom i odabrati update vezu.

10.3. Otvoriti podešavanja procesora – karticu PROPERTIES.

10.4. Odabrati konekciju ka Oracle bazi podataka u kojoj se nalazi Data Mart.

10.5. Postaviti Support Fragmented Transactions na false.

10.6. U polje SQL Statement uneti naredbu koja ažurira postojeću aktivnu torku za proizvod u dimenziji i postavlja vrednost kolone DATE_TO na tekući datum:

```
UPDATE PRODUCT_DIM  
SET DATE_TO = '${now():format('dd-MMM-yy')}'  
WHERE PRODUCT_DIM_ID = ${PRODUCT_DIM_ID}
```

10.7. Dodati PutSQL procesor koji će biti korišćen za unošenje nove torke za proizvod sa ažuriranim informacijama.

10.8. Procesor PutSQL koji ažurira postojeću torku u dimenziji (korak 10.1) povezati sa PutSQL procesorom koji dodaje novu torku za proizvod (korak 10.7) i odabrati success vezu.

10.9. Odabrati konekciju ka Oracle bazi podataka u kojoj se nalazi Data Mart.

10.10. Postaviti Support Fragmented Transactions na false.

10.11. U polje SQL Statement uneti naredbu za dodavanje korisnika sa ažuriranim informacijama:

```
INSERT INTO PRODUCT_DIM (DATE_FROM, DATE_TO, PRODUCT_ID, PRODUCT_NAME,  
LANGUAGE_ID, MIN_PRICE, LIST_PRICE, SUPPLIER_ID, WARRANTY_PERIOD,  
WEIGHT_CLASS, CATEGORY_ID, PARENT_CATEGORY_ID, CATEGORY_NAME)  
VALUES ('${now():format('dd-MMM-yy')}', NULL, ${PRODUCT_ID},  
 '${PRODUCT_NAME}', '${LANGUAGE_ID}', '${MIN_PRICE}', '${LIST_PRICE}',  
 '${SUPPLIER_ID}', '${WARRANTY_PERIOD}', '${WEIGHT_CLASS}',  
 '${CATEGORY_ID}', '${PARENT_CATEGORY_ID}', '${CATEGORY_NAME}')
```

11. Dodavanje LogAttribute procesora

11.1. Dodati LogAttribute procesor.

11.2. Procesor RouteOnAttribute povezati sa LogAttribute procesorom i odabrati no_change vezu.


```
SELECT ORDER_ID, to_char(ORDER_DATE, 'yyyy-mm-dd') AS ORDER_DATE,
CUSTOMER_ID, PRODUCT_ID, QUANTITY, QUANTITY * UNITE_PRICE AS
TOTAL_VALUE FROM ORDERS
```

2. Ovaj procesor se dalje povezuje success vezom sa ConvertAvroToJson procesorom kako bi se izvršilo prebacivanje sadržaja FlowFile-a iz Avro u JSON format.
3. Nakon toga, ConvertAvroToJson procesor se success vezom povezuje sa SplitJson procesorom u okviru koga se svaki JSON objekat FlowFile-a razdvaja u pojedinačan FlowFile.
4. SplitJson procesor se split vezom povezuje sa EvaluateJsonPath procesorom u okviru koga se kreiraju novi atributi FlowFile-a u kojima se pamte vrednosti iz JSON objekta.

Configure Processor | EvaluateJsonPath 1.19.1

Stopped

SETTINGS | SCHEDULING | **PROPERTIES** | RELATIONSHIPS | COMMENTS

Required field

Property	Value
Destination	flowfile-attribute
Return Type	auto-detect
Path Not Found Behavior	ignore
Null Value Representation	empty string
CUSTOMER_ID	\$.CUSTOMER_ID
ORDER_DATE	\$.ORDER_DATE
ORDER_ID	\$.ORDER_ID
PRODUCT_ID	\$.PRODUCT_ID
QUANTITY	\$.QUANTITY
TOTAL_VALUE	\$.TOTAL_VALUE

CANCEL APPLY

5. Za svaki FlowFile potrebno je pročitati vrednosti za atribut tabele činjenica CUSTOMER_DIM_ID, PRODUCT_DIM_ID i DATE_DIM_ID na osnovu vrednosti CUSTOMER_ID, PRODUCT_ID i ORDER_DATE.

5.1. Iz tog razloga, uključujemo ExecuteSQL procesor koji na osnovu vrednosti CUSTOMER_ID čita vrednost CUSTOMER_DIM_ID iz dimenzije korisnika:

```
SELECT CUSTOMER_DIM_ID
FROM CUSTOMER_DIM
WHERE CUSTOMER_ID = ${CUSTOMER_ID} AND DATE_TO IS NULL
```

5.2. Izlaz ovog procesora konvertujemo iz Avro formata u JSON format

5.3. Nakon toga, korišćenjem EvaluateJsonPath procesora dodajemo novi atribut FlowFile-u CUSTOMER_DIM_ID: \$.CUSTOMER_DIM_ID

5.4. Korake 5.1, 5.2 i 5.3 ponavljamo kako bi pročitali vrednosti za PRODUCT_DIM_ID i DATE_DIM_ID

- 5.5. Uključujemo ExecuteSQL procesor koji na osnovu vrednosti PRODUCT_ID čita vrednost PRODUCT_DIM_ID iz dimenzije korisnika:

```
SELECT PRODUCT_DIM_ID
FROM PRODUCT_DIM
WHERE PRODUCT_ID = ${PRODUCT_ID} AND DATE_TO IS NULL
```

- 5.6. Izlaz ovog procesora konvertujemo iz Avro formata u JSON format

- 5.7. Nakon toga, korišćenjem EvaluateJsonPath procesora dodajemo novi atribut FlowFile-u PRODUCT_DIM_ID: \$. PRODUCT_DIM_ID

- 5.8. Uključujemo ExecuteSQL procesor koji na osnovu vrednosti ORDER_DATE čita vrednost DATE_DIM_ID iz dimenzije korisnika:

```
SELECT DATE_DIM_ID
FROM DATE_DIM
WHERE SALES_DATE = TO_DATE('${ORDER_DATE}', 'yyyy-MM-dd')
```

- 5.9. Izlaz ovog procesora konvertujemo iz Avro formata u JSON format

- 5.10. Nakon toga, korišćenjem EvaluateJsonPath procesora dodajemo novi atribut FlowFile-u

- 5.11. DATE_DIM_ID: \$. DATE_DIM_ID

6. Nakon pripreme svih informacija, može da se izvrši upit u tabelu činjenica. Koristi se PutSQL procesor. U podešavanjima ovog procesora definiše se naredba koju procesor treba da izvrši:
- ```
INSERT INTO SALES_FACT VALUES (${ORDER_ID}, ${CUSTOMER_DIM_ID},
${PRODUCT_DIM_ID}, ${DATE_DIM_ID}, ${TOTAL_VALUE}, ${QUANTITY})
```

Configure Processor | PutSQL 1.19.1

Stopped

SETTINGS

SCHEDULING

PROPERTIES

RELATIONSHIPS

COMMENTS

Required field

| Property                        | Value                                                   |
|---------------------------------|---------------------------------------------------------|
| JDBC Connection Pool            | Oracle                                                  |
| SQL Statement                   | INSERT INTO SALES_FACT_2 VALUES (\${ORDER_ID}, \${CU... |
| Support Fragmented Transactions | true                                                    |
| Database Session AutoCommit     | false                                                   |
| Transaction Timeout             | No value set                                            |
| Batch Size                      | 100                                                     |
| Obtain Generated Keys           | false                                                   |
| Rollback On Failure             | false                                                   |

CANCEL

APPLY