

ETL con la Suite Pentaho: *my first data warehouse system*



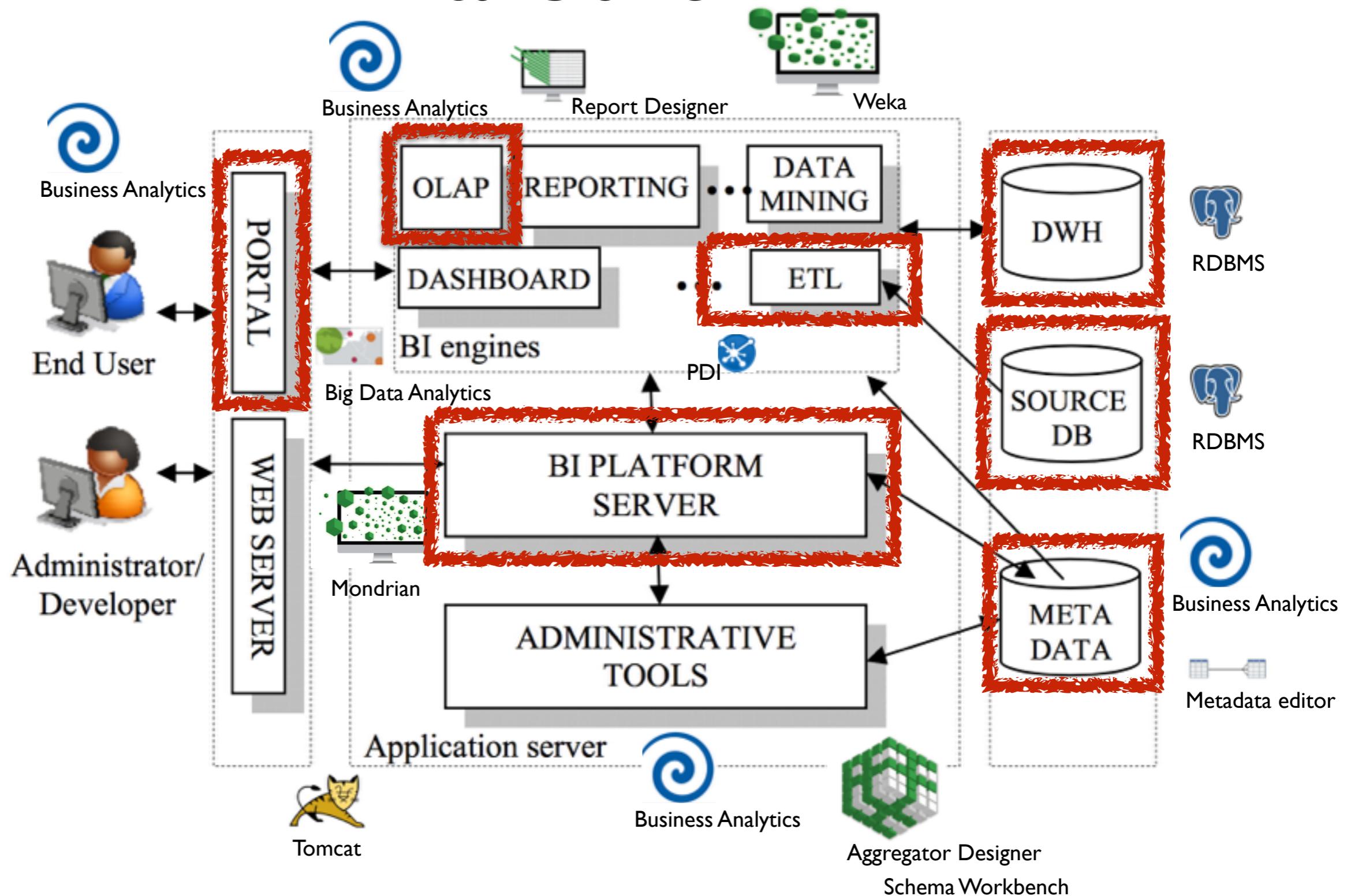
Luigi Bellomarini
luigi.bellomarini@uniroma3.it



La suite Pentaho



La suite



OLAP con Pentaho

- Breve guida all'installazione di:
 - Pentaho BI Server
 - Buisiness Analytics 
 - Mondrian 
 - Data Integration 
 - Utilizzando un RDBMS 
PostgreSQL

Costruzione e utilizzo di un data mart

ETL



Database OLTP

Ricette(Numero, CodFarmacia, CFPaziente, Data)

Farmacie(CodFarmacia, Nome, Via, NumeroCivico, Città)

ElementiRicetta(NumeroRicetta, CodFarmaco, Quantità)

Farmaci(Codice, Descrizione, CodMolecola, CodCasa, Prezzo, Fascia)

Molecole(CodMolecola, Descrizione)

Pazienti(CF, Cognome, Nome, DataNascita, Via, NumeroCivico, Città)

CaseFarmaceutiche(CodCasa, Nome)

ASL(Codice, Nome)

Territorio(Via, Città, NumeroCivico, ASL)

- Ovvi vincoli di integrità referenziale.
- Prezzi, fasce e indirizzi cambiano nel tempo.

Obiettivi

- Costruire un data mart in Pentaho, con uno schema a stella per poter effettuare alcune analisi sulle prescrizioni.
- Costruire un flusso di ETL che alimenti il data mart a partire dal database OLTP.

Costruzione del database OLTP

The screenshot shows the pgAdmin III interface. The left pane is the Object browser, displaying a tree structure of database objects. The 'ElementiRicetta' table is selected, highlighted with a blue border. The right pane contains four tabs: Properties, Statistics, Dependencies, and Dependents. The Properties tab is active, showing detailed information about the table. The SQL pane at the bottom shows the CREATE TABLE statement for 'ElementiRicetta'.

Properties Tab Data:

Property	Value
Name	ElementiRicetta
OID	41239
Owner	postgres
Tablespace	pg_default
ACL	
Of type	
Primary key	NumeroRicetta, CodFarmaco
Rows (estimated)	0
Fill factor	
Rows (counted)	0
Inherits tables	No
Inherited tables count	0
Unlogged?	No
Has OIDs?	No
System table?	No
Comment	

SQL pane:

```
-- Table: "ElementiRicetta"
-- DROP TABLE "ElementiRicetta";

CREATE TABLE "ElementiRicetta"
(
    "NumeroRicetta" integer NOT NULL,
    "CodFarmaco" integer NOT NULL,
    "Quantita" integer,
    CONSTRAINT "ElementiRicetta_pkey" PRIMARY KEY ("NumeroRicetta", "CodFarmaco"),
    CONSTRAINT "ElementiRicetta_CodFarmaco_fkey" FOREIGN KEY ("CodFarmaco")
)
```

DDL

```
CREATE TABLE "ElementiRicetta"
(
    "NumeroRicetta" integer NOT NULL,
    "CodFarmaco" integer NOT NULL,
    "Quantita" integer,
    CONSTRAINT "ElementiRicetta_pkey" PRIMARY KEY ("NumeroRicetta", "CodFarmaco"),
    CONSTRAINT "ElementiRicetta_CodFarmaco_fkey" FOREIGN KEY ("CodFarmaco")
        REFERENCES "Farmaci" ("Codice") MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT "ElementiRicetta_NumeroRicetta_fkey" FOREIGN KEY ("NumeroRicetta")
        REFERENCES "Ricette" ("Numero") MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
);
```

• • •

ASL

Navicat Premium

The screenshot shows the Navicat Premium interface. On the left, the object browser displays a database structure with schemas like MySql5, postgres, farmacie, and public. The public schema contains tables such as ASL, CaseFarmaceutiche, ElementiRicetta, Farmaci, Farmacie, Molecole, Pazienti, Ricette, and Territorio. The ASL table is selected and shown in the main grid view. The grid view displays columns: Codice and Nome. The data shows 8 rows with Codice values from 1 to 8 and Nome values Roma A through Roma H. Below the grid, there is an SQL insert statement: `insert into "public"."ASL" ("Nome", "Codice") values ('Roma H', '8')`. The bottom status bar indicates "1 of 9 selected".

Name	Group	OID	Owner	ACL	Primary Key	Rows	Inherits Table	Inherited Tables Count
ASL		41276	postgres		ASL_pkey	0 NO	0	0

Grid View Form View Image Text Hex Filter Wizard Import Wizard Export Wizard Sort Ascending

insert into "public"."ASL" ("Nome", "Codice") values ('Roma H', '8')

1 of 9 selected

CaseFarmaceutiche

The screenshot shows the Navicat Premium interface connected to the 'CaseFarmaceutiche @farmacie.public (postgres)' database. The left sidebar displays the database structure, including the 'CaseFarmaceutiche' schema which contains tables like ASL, Casel, ElementiRicetta, Farmaci, Farmacie, Molecole, Pazienti, Ricette, and Territorio. The main pane shows the 'CaseFarmaceutiche' table with 12 rows of data:

CodCasa	Nome
1	Menarini
2	ADICA
3	AvantiGarde
4	Biopharma
5	Cosmo
6	Dipharma
7	Eurosital
8	Farma Uno
9	Hikma
10	Iodosan
11	Italchimici
12	Mipharm

The bottom status bar indicates '1 of 9 selected'. The footer shows page numbers 23 and 24.

ElementiRicetta

The screenshot shows the Navicat Premium interface. On the left, the connection tree displays databases like MySQL5, postgres, dwh_ava, dwh_test, and farmacie, with the public schema selected. In the center, the main window shows the 'ElementiRicetta' table from the 'farmacie.public' database. The table has three columns: NumeroRicetta, CodFarmaco, and Quantita. The data grid contains 21 records. Below the grid is a SQL query: 'select * from "public"."ElementiRicetta" limit 1000 offset 0'. At the bottom, there are navigation buttons and a status bar indicating '1 of 9 selected'. The top menu bar includes options like Grid View, Form View, Image, Text, Hex, Filter Wizard, Import Wizard, Export Wizard, and Sort Ascending.

NumeroRicetta	CodFarmaco	Quantita
1	1	1
1	2	3
2	4	5
2	1	1
2	3	2
3	4	1
3	5	5
4	5	2
4	19	1
4	7	1
5	6	1
5	8	3
5	3	2
6	4	10
6	2	2
7	8	1
7	9	1
8	2	2
9	5	2
9	6	1

Farmaci

The screenshot shows the Navicat Premium interface connected to a PostgreSQL database named 'Farmaci @farmacie.public (postgres)'. The left sidebar displays a tree view of the database schema, including tables like ASL, Case, Elementi, Farmaci, Molecole, Pazienti, Ricette, and Territorio. The 'Farmaci' table is selected and highlighted in the tree.

The main window shows a grid view of the 'Farmaci' table with the following data:

N.	Codice	Descrizione	CodMolecola	CodCasa	Prezzo	Fascia
1	ADCIIRCA	1	2	3	15	B
2	AMMONAPS	5	2	14	12	B
3	ANCOTIL	5	2	12	12	B
4	ATRIANCE	4	4	23	12	B
5	FERTAVID	4	4	12	12	B
6	FOLOTYN	4	4	23	12	C
7	MABTHERA	6	2	12	12	B
8	MINIRIN	8	12	12	12	B
9	MYALEPT	5	4	23	12	C
19	VENTAVIS	5	4	12	12	C

Below the table, a SQL query is displayed: `select * from "public"."Farmaci" limit 1000 offset 0`. The results pane shows '10 records in page 1'.

Farmacie

The screenshot shows the Navicat Premium interface connected to the 'farmacie' database. The left sidebar lists databases like MySQL5, postgres, and farmacie, with the 'farmacie' database expanded to show its schemas: public, dwh_ava, dwh_test, and others. The 'public' schema is selected. The main pane displays the 'Farmacie' table in Grid View, showing 15 records. The columns are CodFarmacia, Nome, via, NumeroCivico, and Citta. The records list various pharmacy locations in Rome. Below the grid, a SQL query is shown: 'select * from "public"."Farmacie" limit 1000 offset 0'. The bottom status bar indicates '1 of 9 selected'.

CodFarmacia	Nome	via	NumeroCivico	Citta
1	EUROPEA DR.I AL	Largo Visconti Ver.	12	Roma
2	SANTA RITA SNC	Via Nazionale	13	Roma
3	MADONNA DELL'	Via Candia	134	Roma
4	GIACCONE DI BR.	Via Angelo Emo	120	Roma
5	PIRONA DI PIRON	Corsso Trieste	278	Roma
6	BEATO ANGELICC	Piazza Euclide	22	Roma
7	SANTI COSMA E L	Piazza dei Navigat	12	Roma
8	SANTA SILVIA SNC	Piazza Euclide	16	Roma
9	MIRAFIORI SNC C	Via del Pigneto	99	Roma
10	PITAGORA SNC D	Via di Casal del M.	129	Roma
11	CORSO SIRACUS.	Via Mario Rigamor	200	Roma
12	BORGATA LESNA	Via Conca d'Oro	200	Roma
13	SAN GIOVANNI St	Via dei Castani	23	Roma
14	CAVANNA DELLE	Via della Giuliana	125	Roma
15	MADONNA DELLA	Viale Eritrea	190	Roma

Molecole

The screenshot shows the Navicat Premium interface connected to a PostgreSQL database named 'farmacie'. The left sidebar displays the database structure, including schemas like 'ASL', 'Caselli', 'Elementi', 'Farmaci', 'Farmacie', 'Molecole', 'Pazienti', 'Ricette', and 'Territorio', along with 'Views', 'Functions', 'Queries', and 'Backups'. The 'Molecole' table is selected in the 'Tables' section of the sidebar.

The main window displays the contents of the 'Molecole' table in Grid View. The table has two columns: 'CodMolecola' (Number) and 'Descrizione' (Text). The data consists of 14 rows:

CodMolecola	Descrizione
1	Abciximab
2	Acarbosio
3	Bacitracina
4	Benserazide
5	Buformina
6	Buspirone
7	Ganciclovir
8	Glimepride
9	Nafazolina
10	Rabeprazolo
11	Ritodrina
12	Rosiglitazone
13	Ritonavir
14	Ureidopenicilline

Below the table, a SQL query is shown: `select * from "public"."Molecole" limit 1000 offset 0`. The status bar at the bottom indicates "1 of 9 selected".

Pazienti

The screenshot shows the Navicat Premium interface connected to a PostgreSQL database. The left sidebar lists databases (MySQL5, postgres, farmacie), schemas (dwh_ava, dwh_test, public), and tables (ASL, Caselli, Elementi, Farmaci, Farmacie, Molecole, Pazienti, Ricette, Territorio). The 'Pazienti' table is selected and highlighted with a blue rectangle. The main window displays the table's data in a grid view with the following columns: CF, Cognome, Nome, DataNascita, Via, and NumeroCivico. The data consists of 11 rows, each representing a patient record. A search bar at the top right is empty.

CF	Cognome	Nome	DataNascita	Via	NumeroCivico
B	Baroni	Francesca	1967-05-02	Largo Visconti Ven.	12
E	Carbone	Massimo	1987-05-21	Via Nazionale	13
A	De Angelis	Antonio	1988-05-11	Piazza di Spagna	25
M	D'Amato	Maria	2015-05-13	Corso Trieste	278
R	Ferraro	Rosario	2000-05-17	Corso di Francia	13
D	Giovanni	Domenico	2015-05-20	Piazza dei Navigati	12
C	Giannì	Carmela	2015-05-14	Via Cesare Beccaria	22
N	Giannì	Nicola	1934-07-10	Via Mario Rigamonti	200
F	Giannì	François	1997-05-22	Viale Jonio	222
P	Giannì	Patricia	1997-05-27	Via della Giuliana	125
R	Giannì	Rosa	2015-04-04	Viale Eritrea	190

select * from "public"."Pazienti" limit 1000 offset 0

11 records in page 1

Ricette

The screenshot shows the Navicat Premium interface connected to the 'Ricette @farmacie.public (postgres)' database. The left sidebar displays a tree view of the database schema, including MySQL5, postgres, dwh_ava, dwh_test, farmacie, and public. Under the public schema, there are several tables: ASL, Caselli, Elementi, Farmaci, Farmacie, Molecole, Pazienti, Ricette, and Territorio. The 'Ricette' table is selected and highlighted with a blue border. The main pane shows a grid view of the 'Ricette' table with the following data:

N.	Numero	CodFarmacia	Data	CFPaziente
1	1	2015-05-02	AMN	
2	1	2015-04-04	BLA	
3	3	2015-05-08	BNT	
4	4	2015-02-06	DCC	
5	5	2015-03-13	GRC	
6	6	2014-09-06	MCH	
7	7	2014-06-14	RNA	
8	8	2012-07-12	PVN	
9	9	2015-05-15	RCC	

Below the grid, a SQL query is displayed: `select * from "public"."Ricette" limit 1000 offset 0`. The status bar at the bottom indicates "1 of 9 selected".

Territorio

The screenshot shows the Navicat Premium interface for a PostgreSQL database named 'Territorio @farmacie.public (postgres)'. The left sidebar lists databases like MySQL5, postgres, and farmacie, with 'farmacie' expanded to show its 'public' schema containing tables such as ASL, CaseFarmaceutiche, ElementiRicetta, Farmaci, Farmacie, Molecole, Pazienti, Ricette, and Territorio. The 'Territorio' table is selected and highlighted in the list.

The main window displays the 'Territorio' table in Grid View. The columns are labeled: Nome, Via, Città, NumeroCivico, and ASL. The data shows various street names and addresses in Rome, categorized by ASL (Local Health Service). A total of 24 records are shown in the current page.

Nome	Via	Città	NumeroCivico	ASL
ASL	Via Largo Visconti Ven	Roma	12	1
Case	Via Nazionale	Roma	13	1
Elementi	Piazza di Spagna	Roma	25	1
Farmaci	Viale Ippocrate	Roma	234	2
Farmacie	Via Candia	Roma	134	2
Molecole	Corso Trieste	Roma	278	3
Pazienti	Via Angelo Emo	Roma	120	2
Ricette	Via Crescentio	Roma	90	2
Territorio	Corso di Francia	Roma	13	3
	Via Cipro	Roma	13	3
	Piazza Euclide	Roma	22	3
	Via Nemorense	Roma	33	3
	Circonvallazione C	Roma	45	3
	Piazza del Navigat	Roma	12	4
	Piazza Euclide	Roma	16	4
	Via Cesare Beccar	Roma	22	5
	Via del Pigneto	Roma	99	5
	Via dei Castani	Roma	23	5
	Via di Casal del Mi	Roma	129	6
	Via Mario Rigamor	Roma	200	6

Below the table, a SQL query is displayed: `select * from "public"."Territorio" limit 1000 offset 0`. The status bar at the bottom indicates "1 of 9 selected".

Schema dimensionale

- **FattiPrescrizioni**(KData, KFarmaco, KRicetta,
KASLfarmacia, KASLpaziente, KFasciaEta, Quantita,
Importo)
- **Farmaci**(KFarmaco, Codice, Descrizione, CodMolecola,
DescrizioneMolecola, CodCasa, NomeCasa, Fascia)
- **ASL**(KASL, CodiceASL, Nome)
- **FasciaEta**(KFasciaEta, DescrizioneFasciaEta, da, a)
- **Data**(KData, giorno, mese, anno)

Costruzione dello schema dimensionale

The screenshot shows the pgAdmin III interface with the following details:

- Object browser:** On the left, it lists database objects under the schema "fascia".
 - Tables (5):**
 - ASL:** Contains columns KASL, CodiceASL, and Nome.
 - Data:** Contains columns KData, Giorno, mese, and anno.
 - Farmaci:** Contains columns KFarmaco, Codice, Descrizione, CodMolecola, DescrizioneMolecola, CodCasa, NomeCasa, and Fascia.
 - Constraints (1):** Associated with the ASL table.
 - Indexes (0):** Associated with the ASL table.
 - Rules (0):** Associated with the ASL table.
 - Triggers (0):** Associated with the ASL table.
- Properties pane:** Shows properties for the "FasciaEta" table.

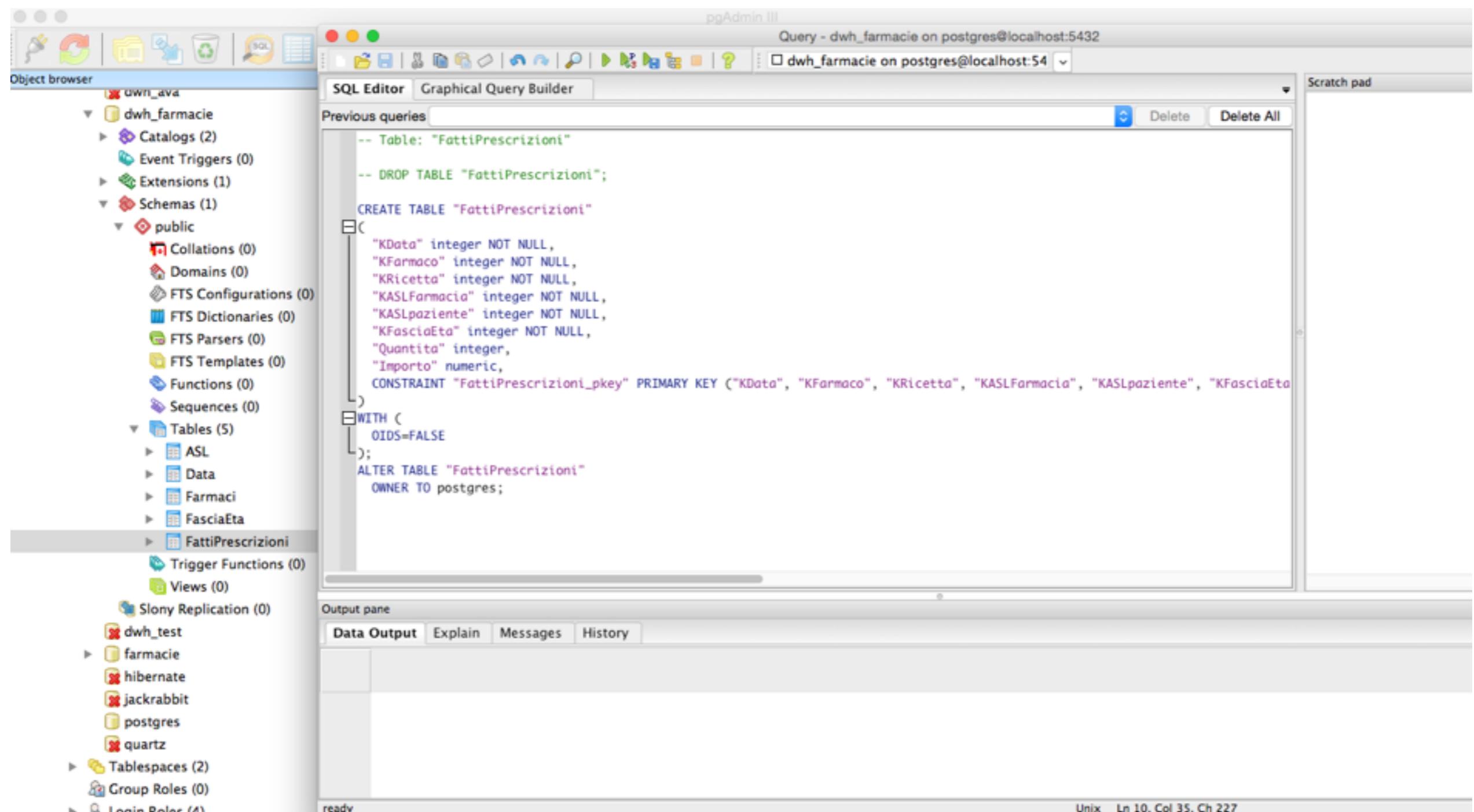
Property	Value
Name	FasciaEta
OID	41354
Owner	postgres
Tablespace	pg_default
ACL	
Of type	
Primary key	KFasciaEta
Rows (estimated)	0
Fill factor	
Rows (counted)	0
Inherits tables	No
Inherited tables count	0
Unlogged?	No
Has OIDs?	No
System table?	No
Comment	
- SQL pane:** Displays the SQL code for creating the "FasciaEta" table.

```
-- Table: "FasciaEta"
-- DROP TABLE "FasciaEta";

CREATE TABLE "FasciaEta"
(
    "KFasciaEta" integer NOT NULL,
    "DescrizioneFasciaEta" character varying,
    CONSTRAINT "FasciaEta_pkey" PRIMARY KEY ("KFasciaEta")
)
WITH (

```

DDL



Alimentazione data mart

The screenshot shows the Pentaho Data Integration (Spoon) application window. The title bar reads "Spoon - Benvenuto!". The left sidebar is titled "Albero principale" and contains "Esploratore" with "Trasformazioni" and "Job" listed. The main area displays the "Benvenuto!" page with the heading "Pentaho Data Integration". A navigation bar at the top of the page includes "Welcome" (highlighted in yellow), "Meet the Family", "Credits", and "Many Reasons to Get Enterprise Edition". Below this, a section titled "Get the Most From Pentaho" features the text "Let us help you become an ETL, Big Data Master." and a "Tutorials & Videos →" link. To the right, there is an illustration of a computer monitor displaying the Spoon interface, with a play button icon overlaid. A circular inset shows a person pointing at a cloud labeled "TRANSFORMAZIONE". At the bottom, a call-to-action button says "Take your ETL, Analytics & Big Data to the Next Level" followed by the text "Want to take your implementation to the next level? Experience for yourself our comprehensive data integration, visualization and analysis tools and create customizable reports and interactive dashboards." and a "Get Pentaho Enterprise Edition" button.

Pentaho data integration

The screenshot shows the Pentaho Data Integration software interface. On the left, there's a sidebar with icons for file operations (New, Open, Save, Print, Import/Export) and navigation (Albero principale, Esploratore, Trasformazioni, Job). The main window title is "Spoon - Benvenuto". A context menu is open over the "Job" item in the sidebar, listing options like Nuovo, Apri, and Connessione database. The main content area displays a "Pentaho Data Integration" welcome page with tabs for Welcome, Meet the Family, Credits, and Many Reasons to Get Enterprise Edition. Below the tabs, there's a section titled "Get the Most From Pentaho" with the subtext "Let us help you become an ETL, Big Data Master." and a "Tutorials & Videos" link. To the right, there's a graphic of a computer monitor displaying the software interface, with a circular inset showing a person pointing at a cloud labeled "TRANSFORMA". At the bottom, a call-to-action button says "Take your ETL, Analytics & Big Data to the Next Level" and "Want to take your implementation to the next level? Experience for yourself our comprehensive data integration, visualization and analysis tools and create customizable reports and interactive dashboards." A blue button at the very bottom reads "Get Pentaho Enterprise Edition".

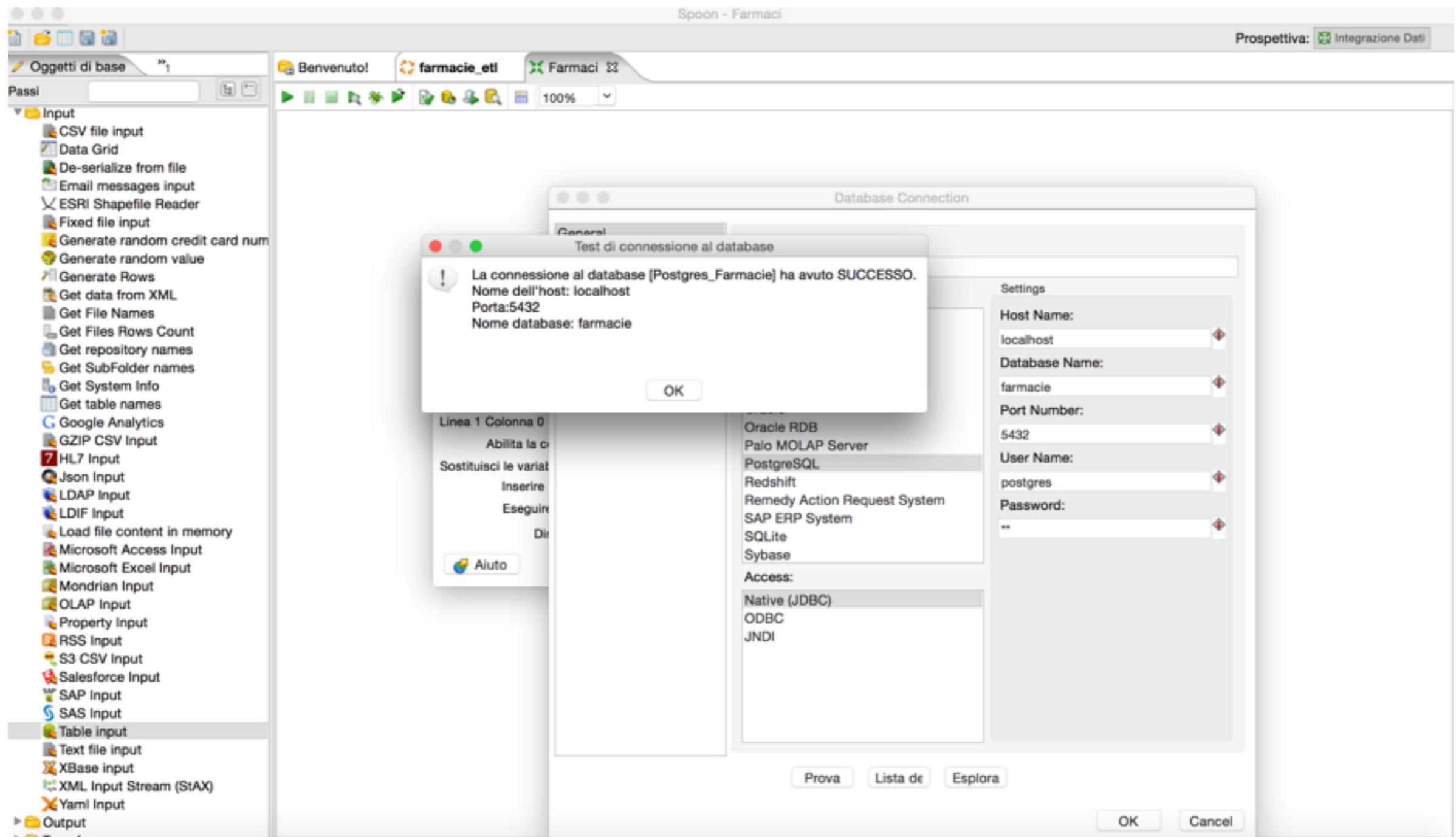
Pentaho data integration

- **Job**
 - Coordinano insiemi di trasformazioni.
 - Permettono il sequenziamento logico/temporale delle trasformazioni.
 - Sono il “main” di un flusso ETL.
 - Gli step sono eseguiti secondo l’ordinamento topologico.
 - Il successivo si avvia dopo la terminazione del precedente.
- **Trasformazioni**
 - Sono l’implementazione vera e propria del flusso di ETL.
 - Gli step sono eseguiti in stream.
 - Tutti gli step sono avviati contemporaneamente e i dati fluiscono.
 - Hanno il controllo delle transazioni.

Progettazione di flussi ETL

- Insieme di pattern e good practices.
- Progettazione logica:
 - **una trasformazione per ogni dimensione.**
 - **una trasformazione per ogni fatto.**
 - **un job che coordina tutte le trasformazioni delle dimensioni.**
 - **un job che alimenta tutte le fact table di un data mart.**
- Procedere bottom-up nei casi semplici (prima le trasformazioni).
- Progettare prima l'impianto iniziale e poi l'update delle dimensioni (e dei fatti).
- Progettazione fisica:
 - Attenzione al dimensionamento delle risorse.
 - Memoria, swapping, redologs, etc ...

Dimensione Farmaci



Farmaci

Spoon - Farmaci (cambiato)

Prospectiva: Integrazione Dati

Oggetti di base *1

Passi

Input

- CSV file input
- Data Grid
- De-serialize from file
- Email messages input
- ESRI Shapefile Reader
- Fixed file input
- Generate random credit card num
- Generate random value
- Generate Rows
- Get data from XML
- Get File Names
- Get Files Rows Count
- Get repository names
- Get SubFolder names
- Get System Info
- Get table names
- Google Analytics
- GZIP CSV Input
- HL7 Input
- Json Input
- LDAP Input
- LDIF Input
- Load file content in memory
- Microsoft Access Input
- Microsoft Excel Input
- Mondrian Input
- OLAP Input
- Property Input
- RSS Input
- S3 CSV Input
- Salesforce Input
- SAP Input
- SAS Input
- Table input
- Text file input
- XBase input
- XML Input Stream (StAX)
- Yaml Input

Output

Benvenuto! farmacie_etl Farmaci

100%

Mantenere le query semplici valutando l'introduzione i join

Tabella d'input

Nome passo Leggi Farmaci

Connessione Postores_Farmacie Modifica... Nuovo... Wizard... Preleva SQL select statement...

SQL

```
SELECT *  
FROM public."Farmaci";
```

Righe di passo: Leggi Farmaci (10 righe)

#	Codice	Descrizione	CodMolecola	CodCasa	Prezzo	Fascia
1	1	ADCIRCA	1	2	3	A
2	2	AMMONAPS	5	2	15	B
3	3	ANCOTIL	5	2	14	B
4	4	ATRIANCE	4	4	12	B
5	5	FERTAVID	4	4	12	B
6	6	FOLOTYN	4	4	23	C
7	7	MABTHERA	6	2	12	B
8	8	MINIRIN	8	12	12	B
9	9	MYALEPT	5	4	23	C
10	19	VENTAVIS	5	4	12	C

Linea 4 Colonna 17

Abilita la conversione lazy

Sostituisci le variabili nello script?

Inserire i dati dal passo

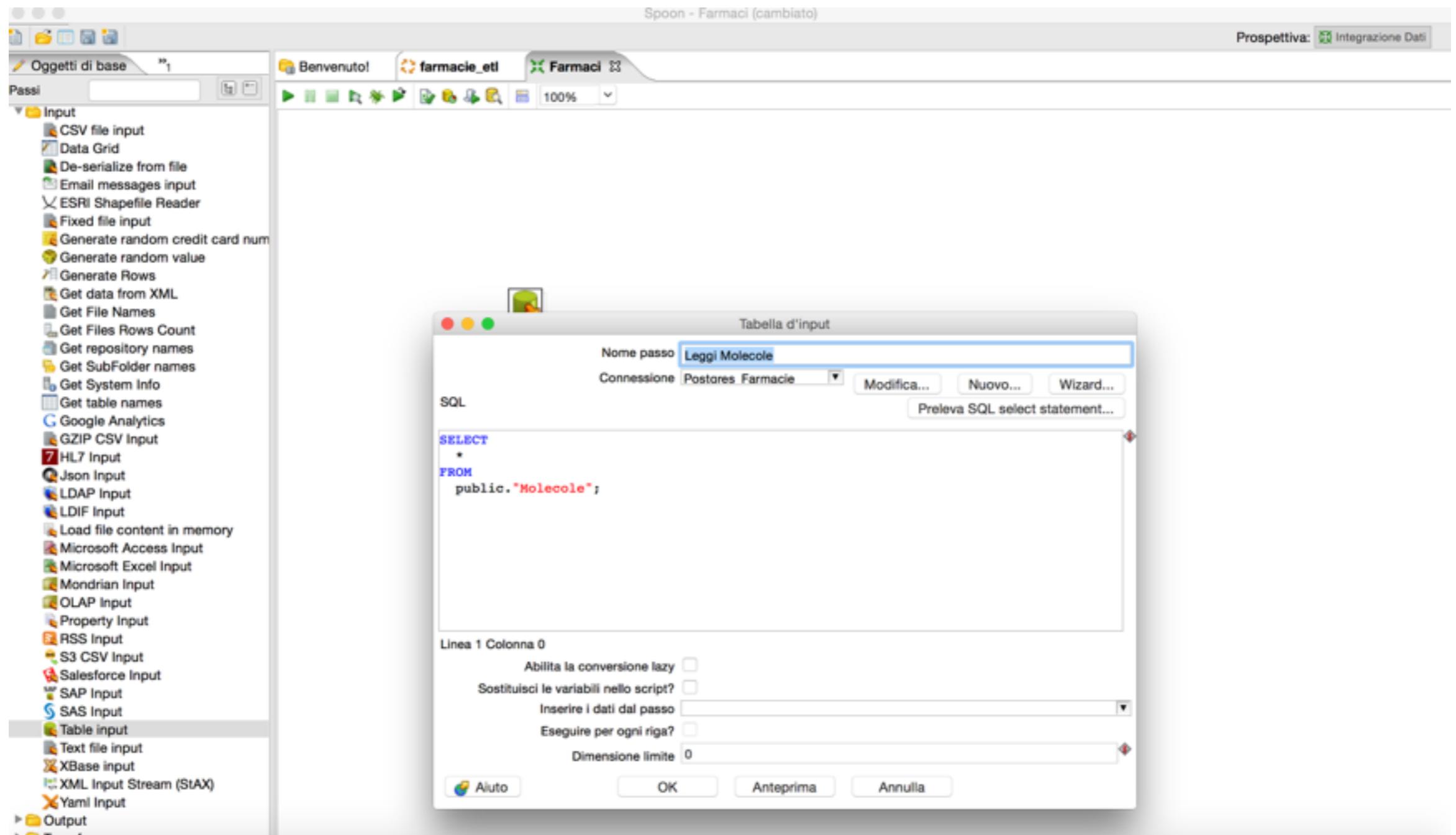
Eseguire per ogni riga?

Dimensione limite

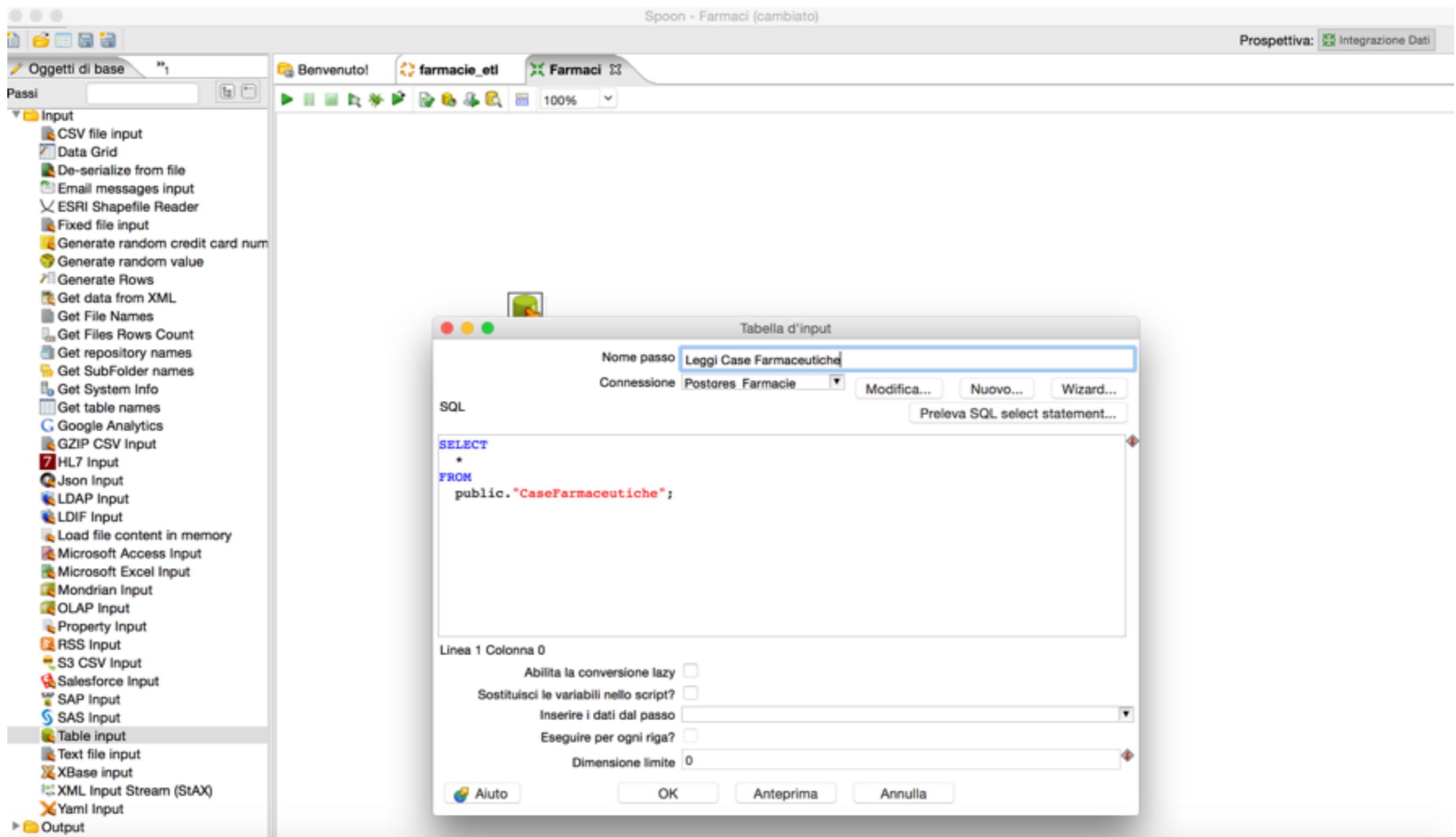
Chiudi Mostra log

OK Anteprima Annulla Aiuto

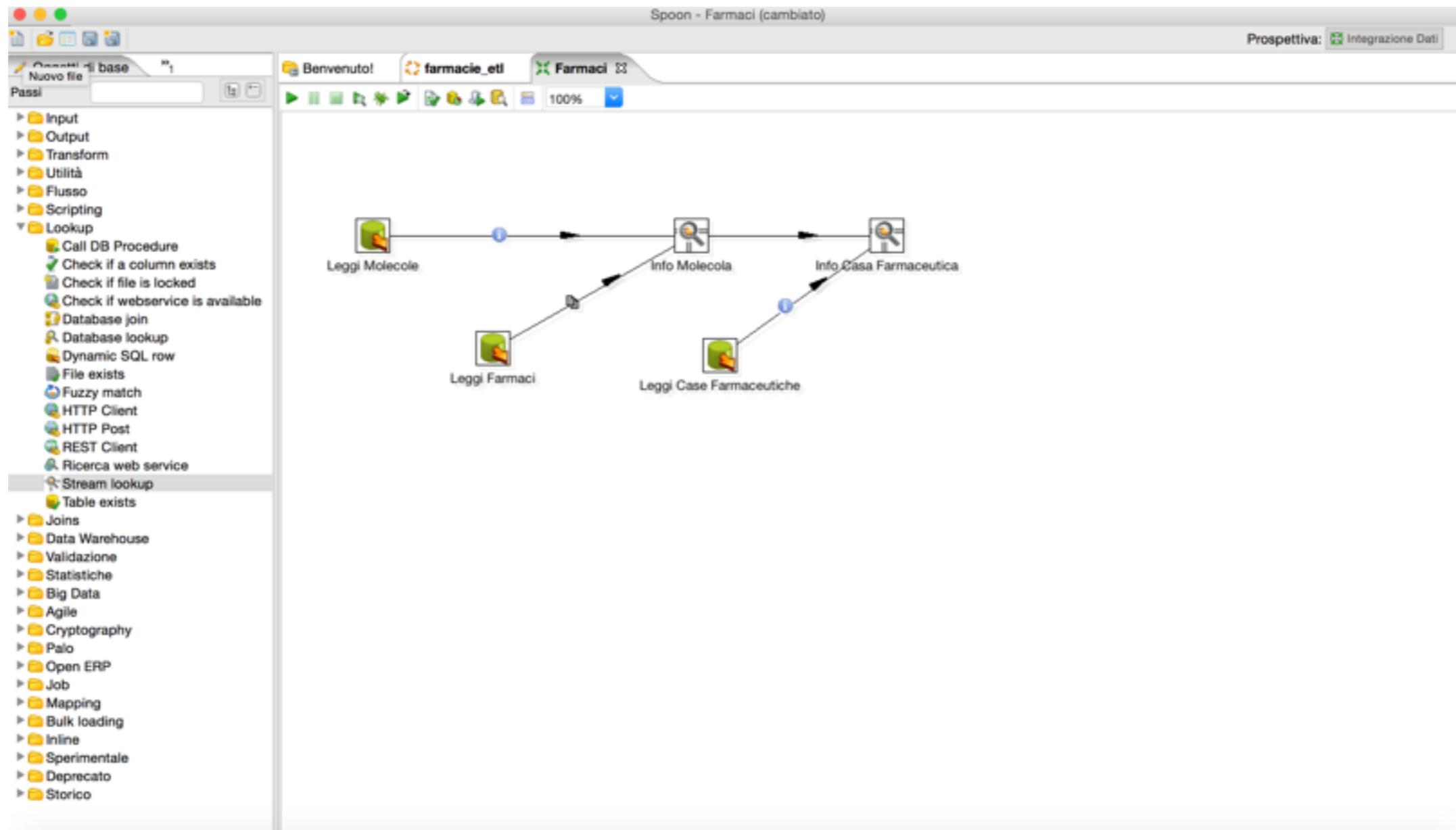
Molecole



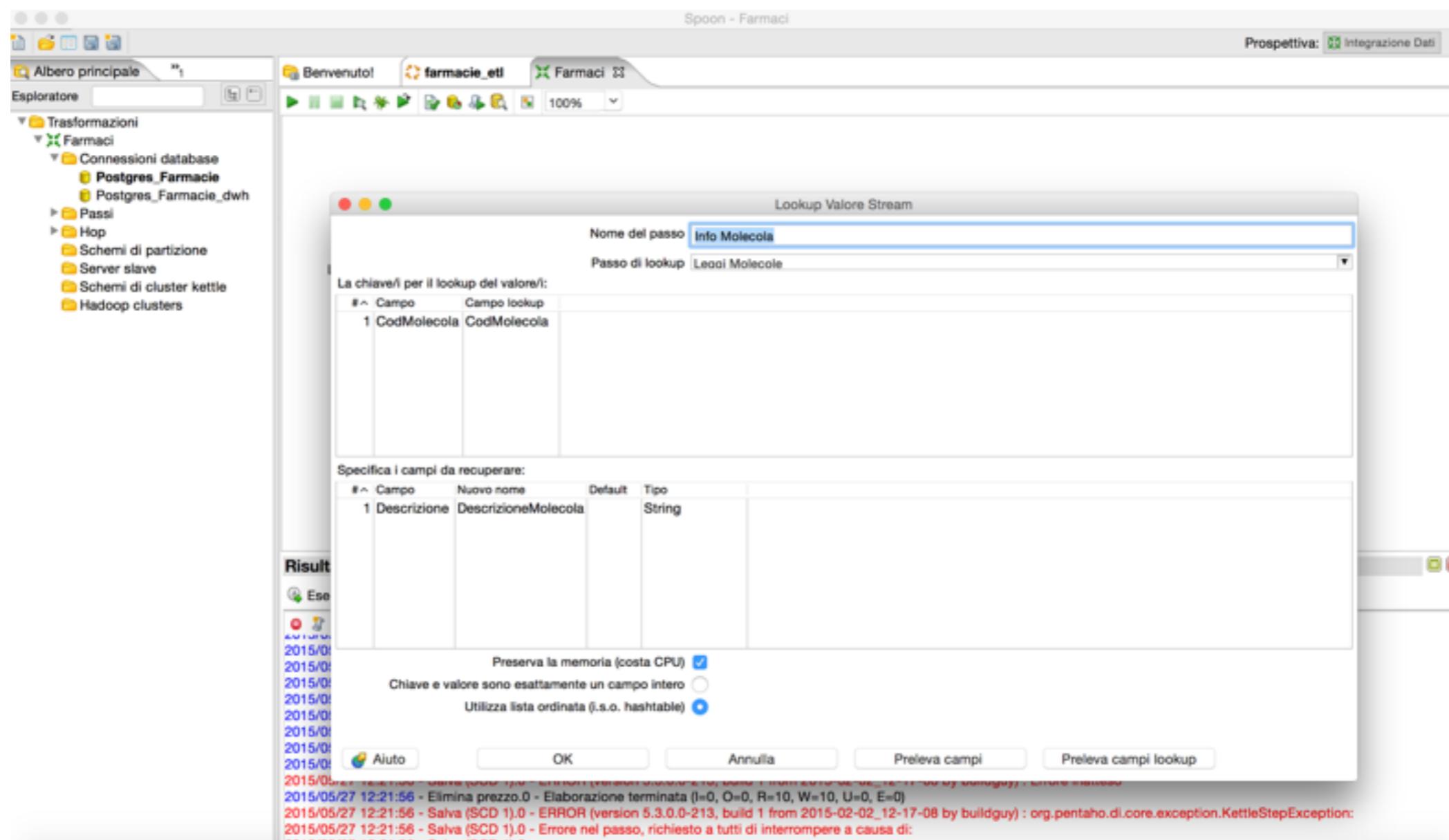
Case farmaceutiche



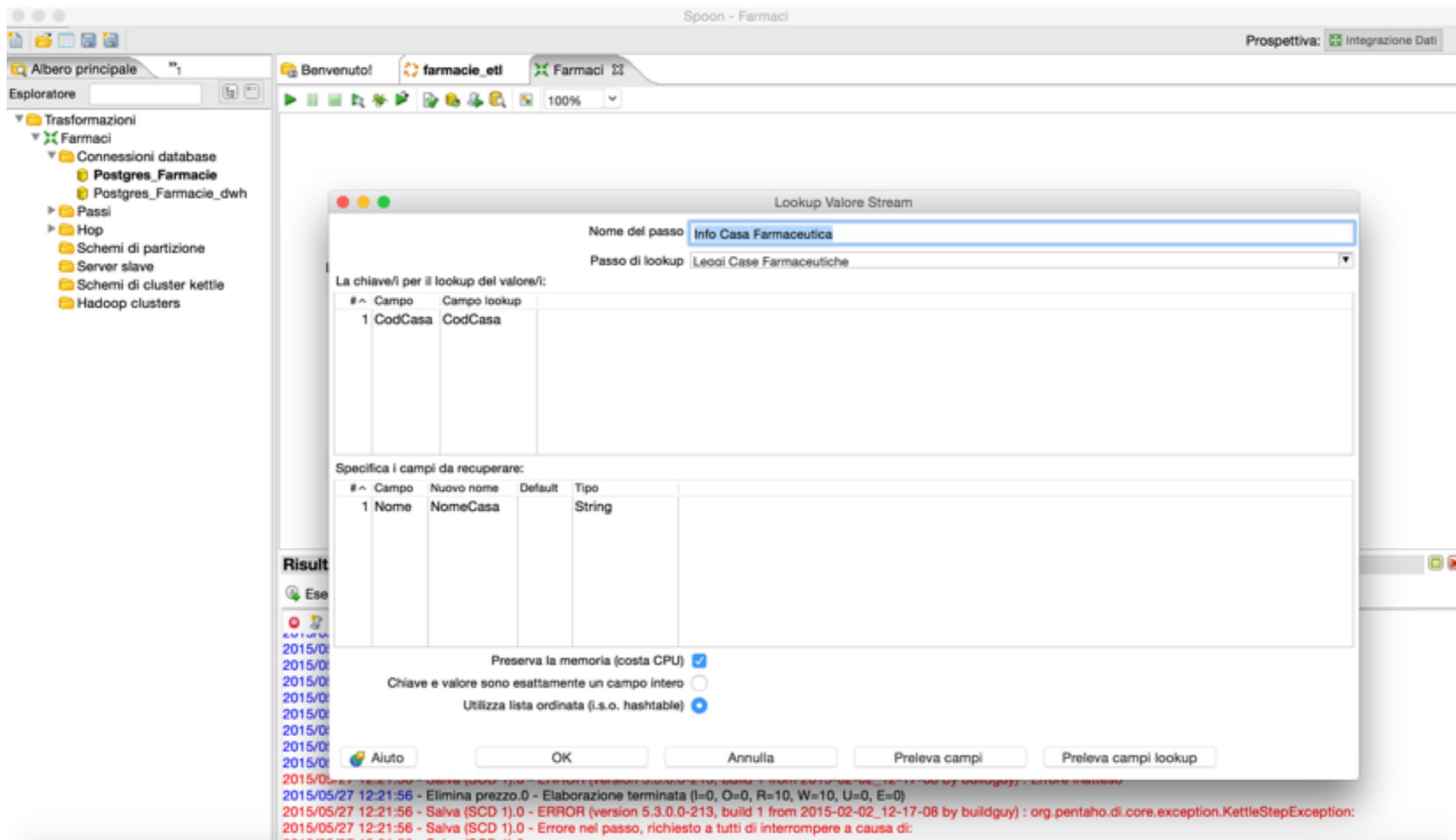
Arricchire gli stream



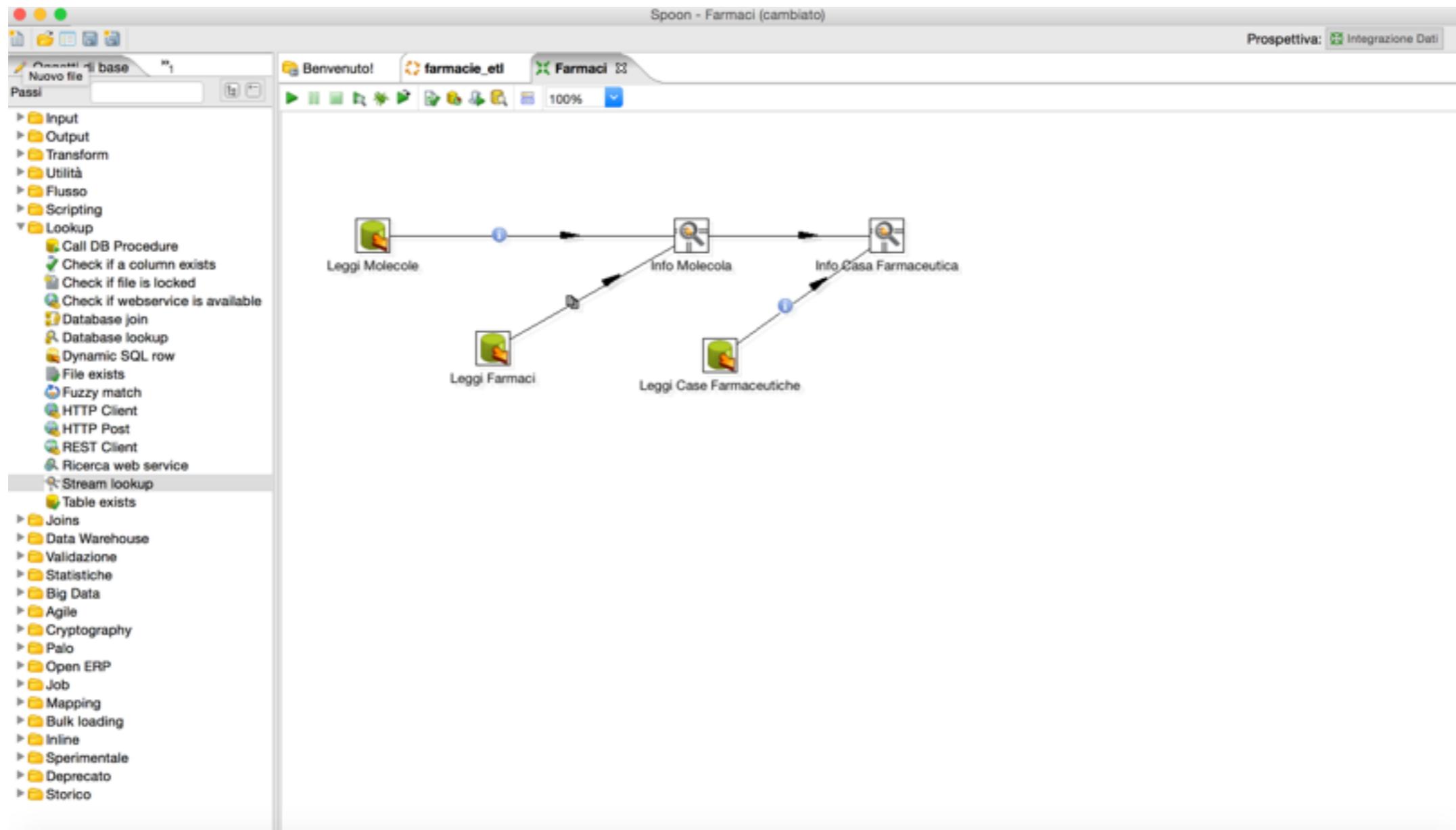
Stream Lookup (Molecola)



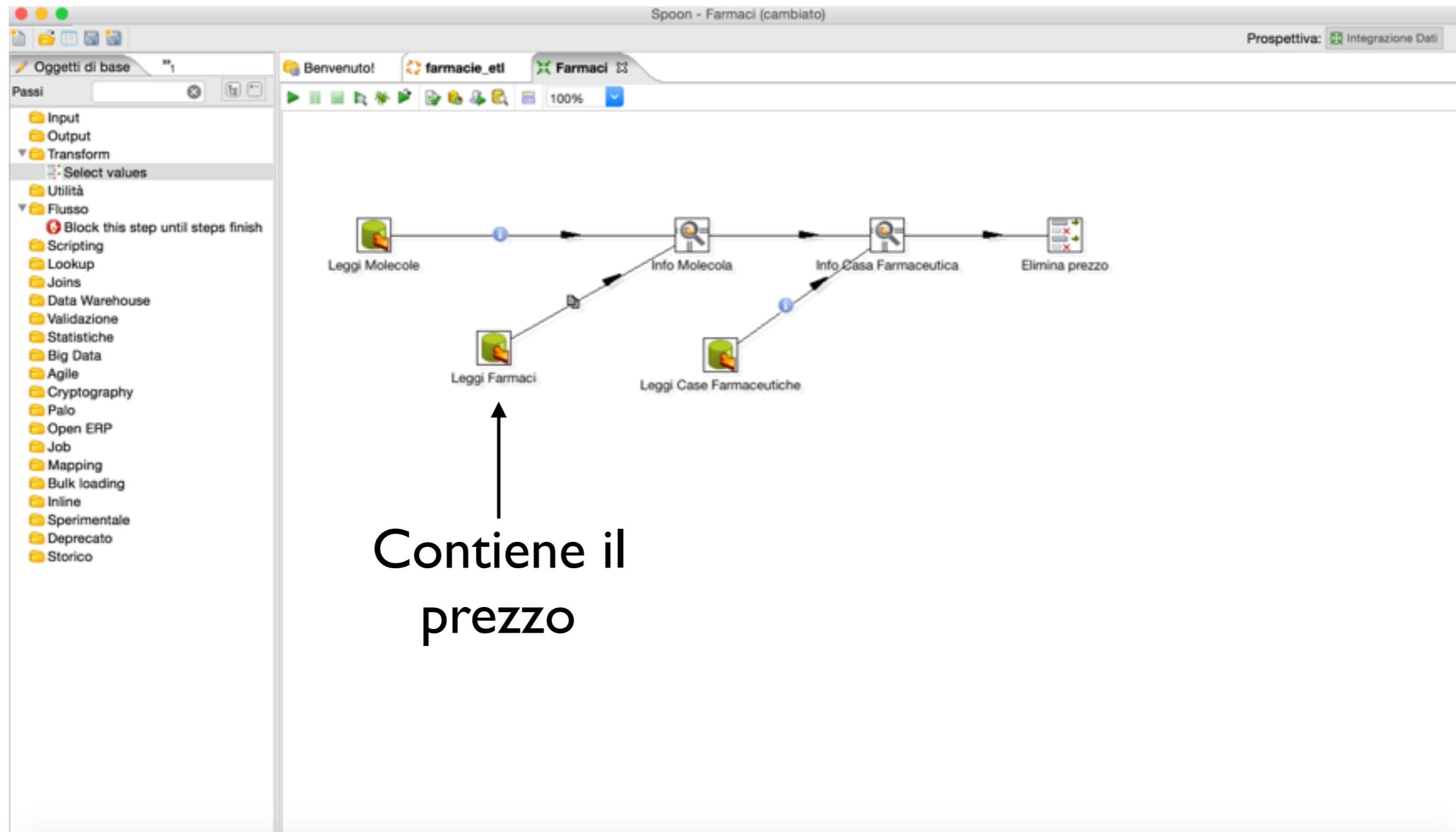
Stream Lookup (Casa Farmaceutica)



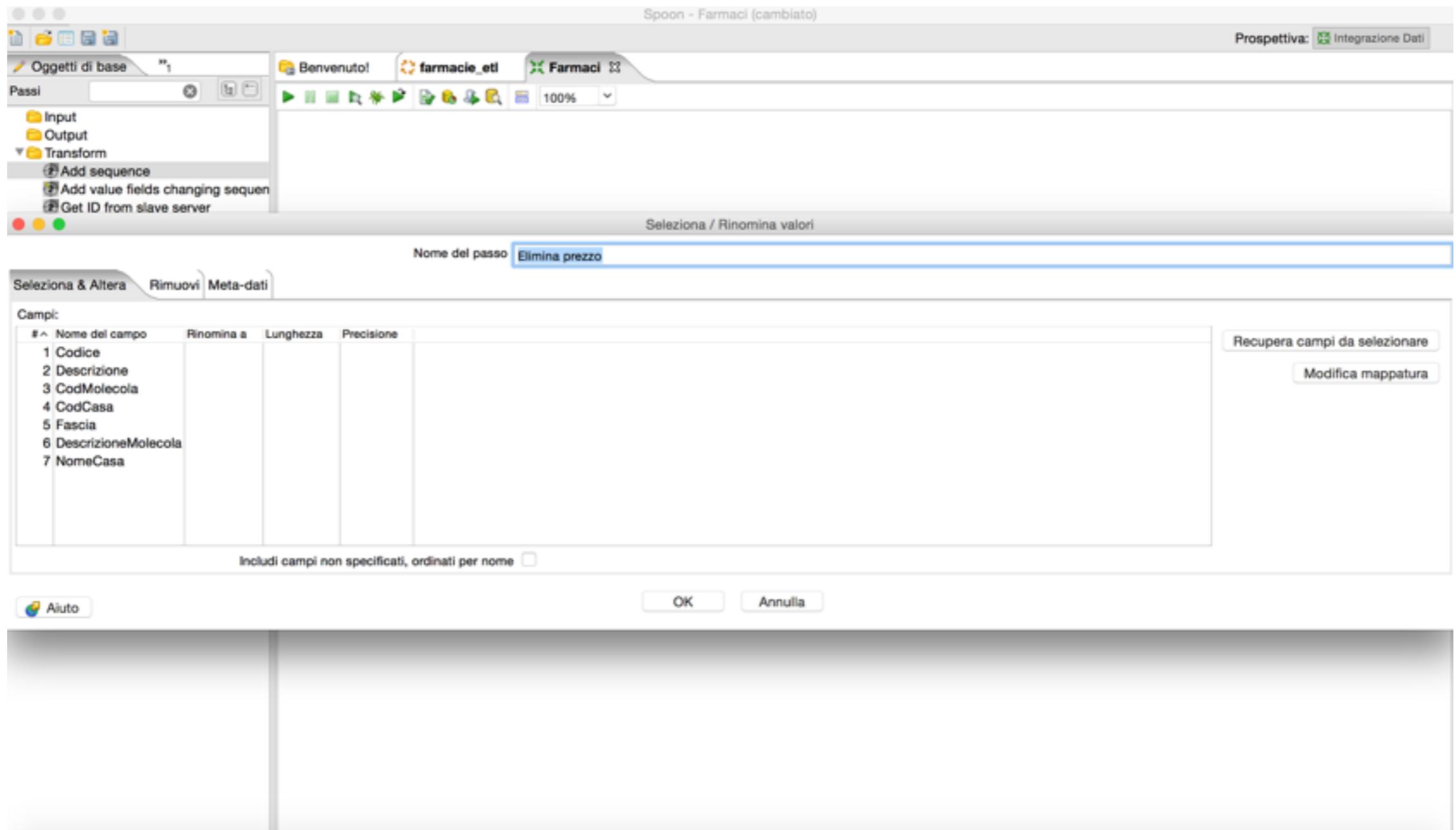
Dove siamo



Selezionare attributi



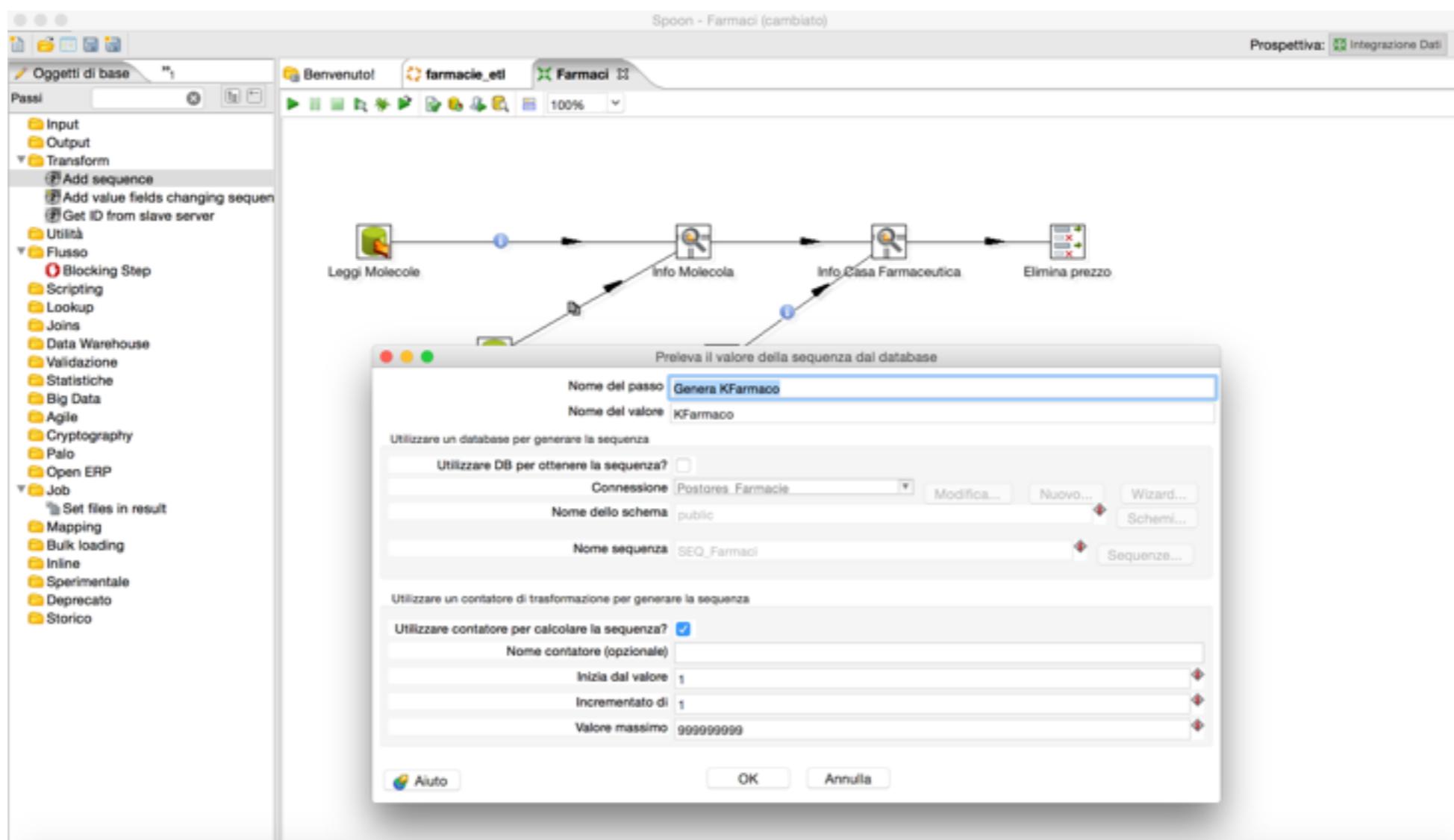
Selezionare attributi



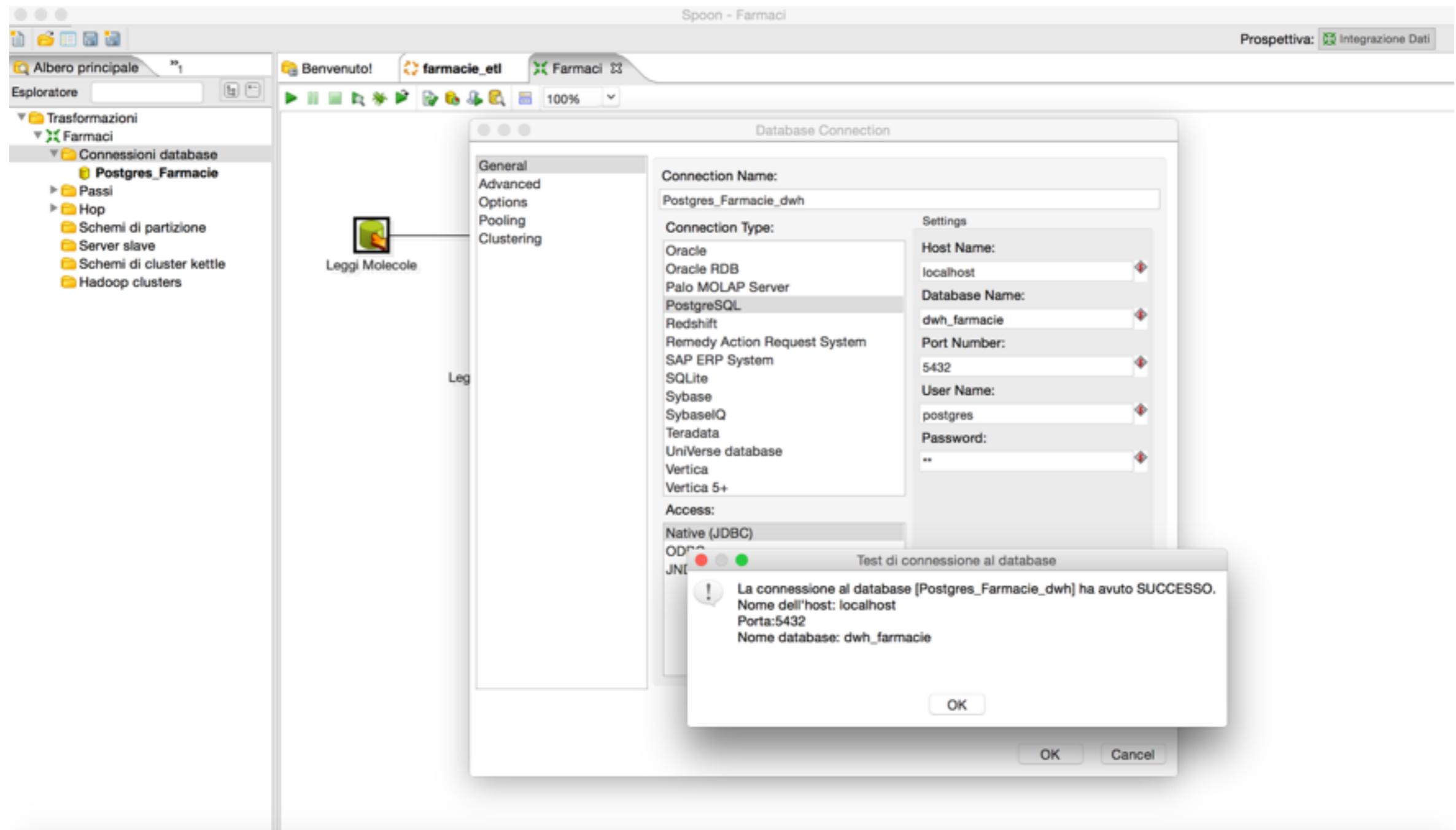
Creare una chiave surrogata

- definire una SEQUENCE:
 - CREATE SEQUENCE public."SEQ_Farmaci";
 - utilizzare un calcolatore ETL.

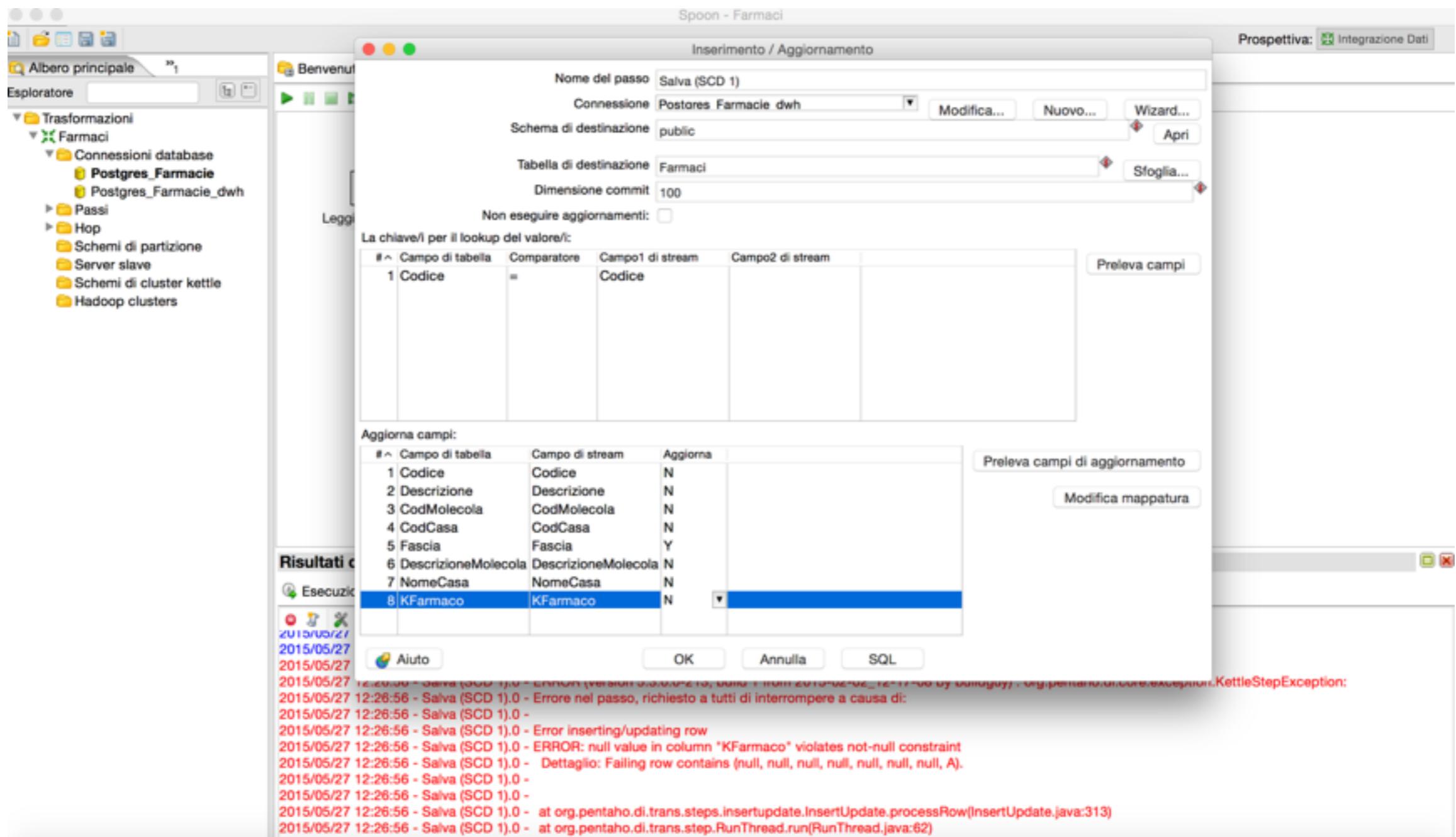
Creare una chiave surrogata



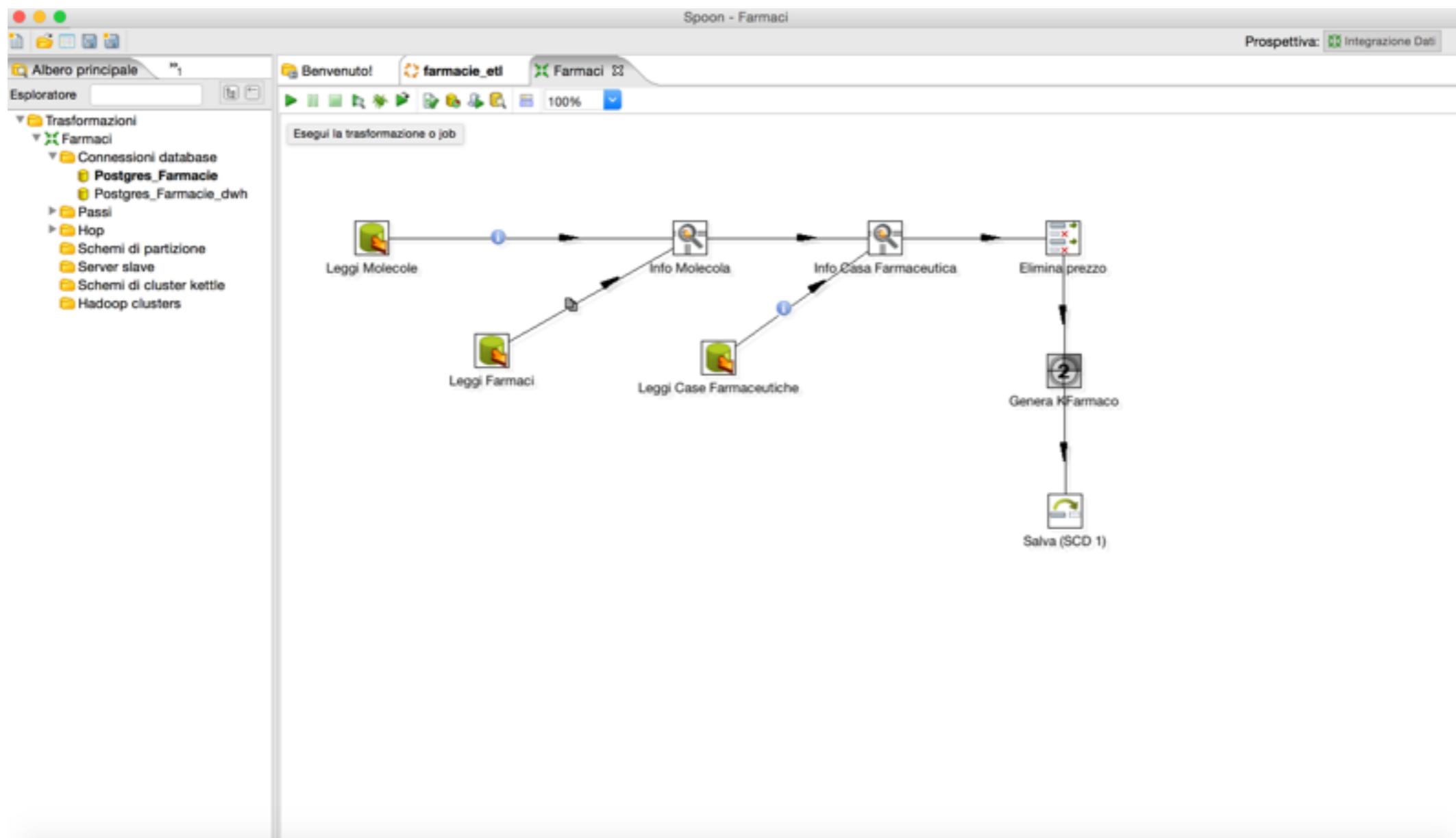
L'output



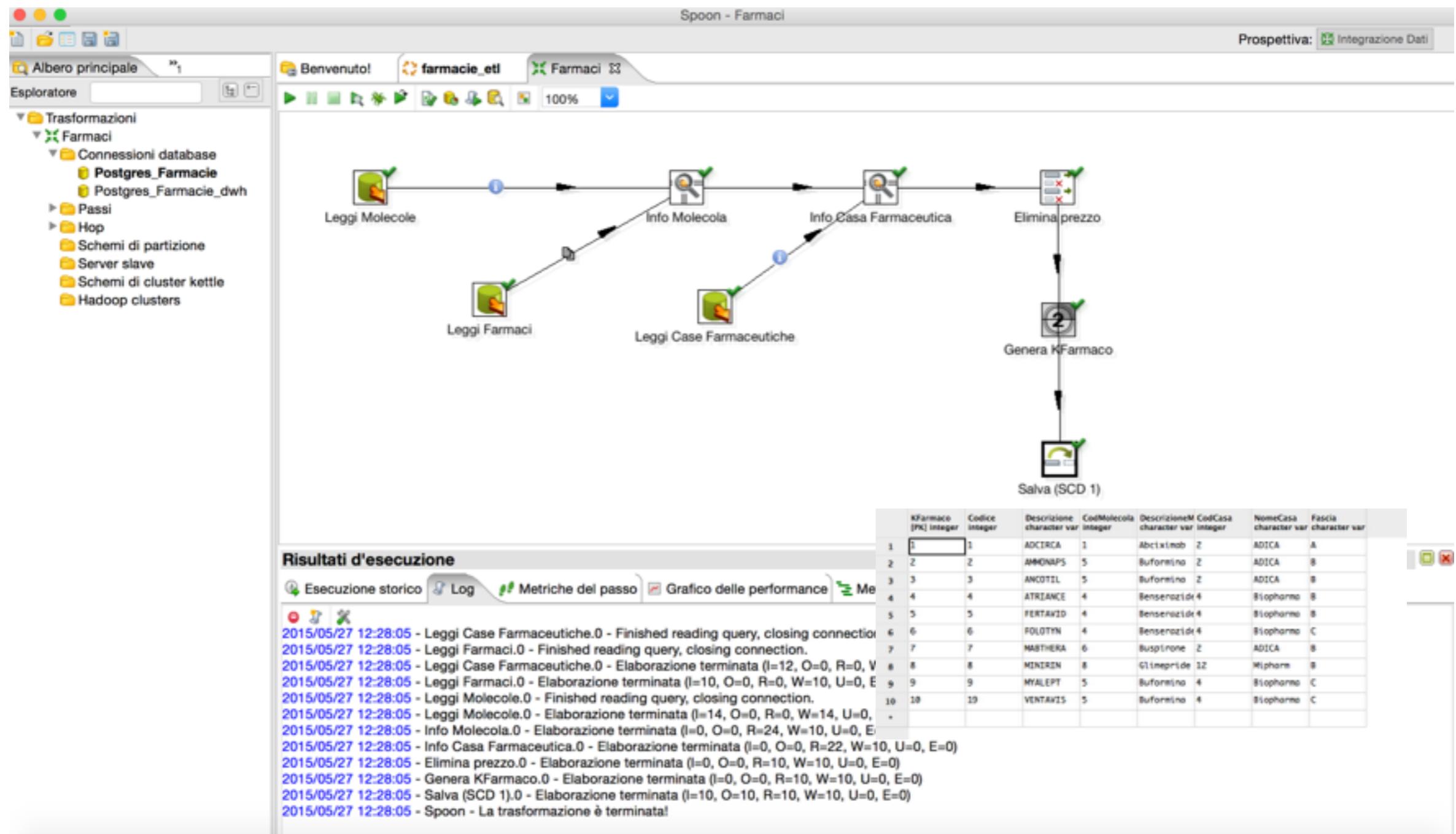
Gestire le variazioni e salvare



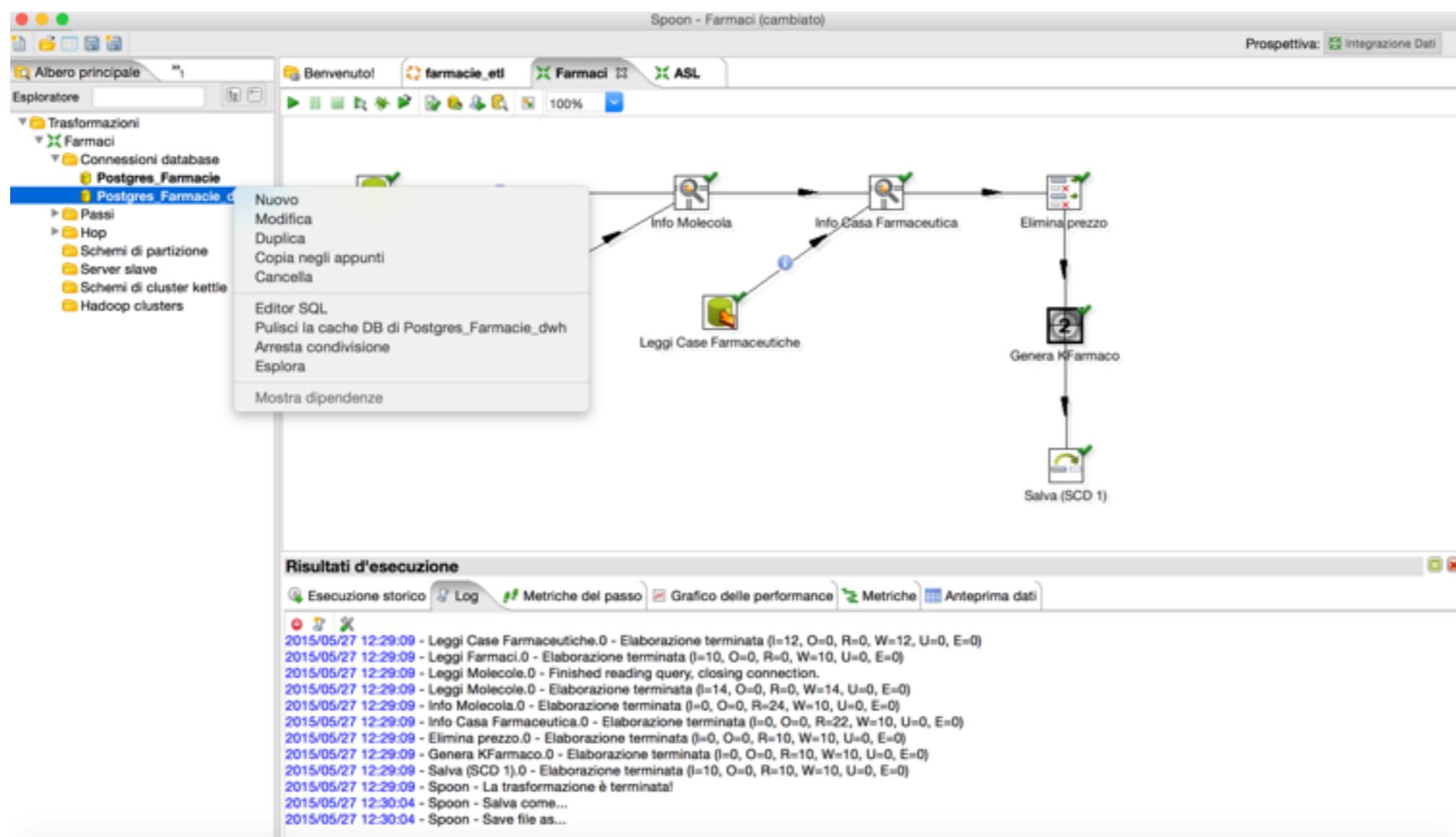
La trasformazione completa



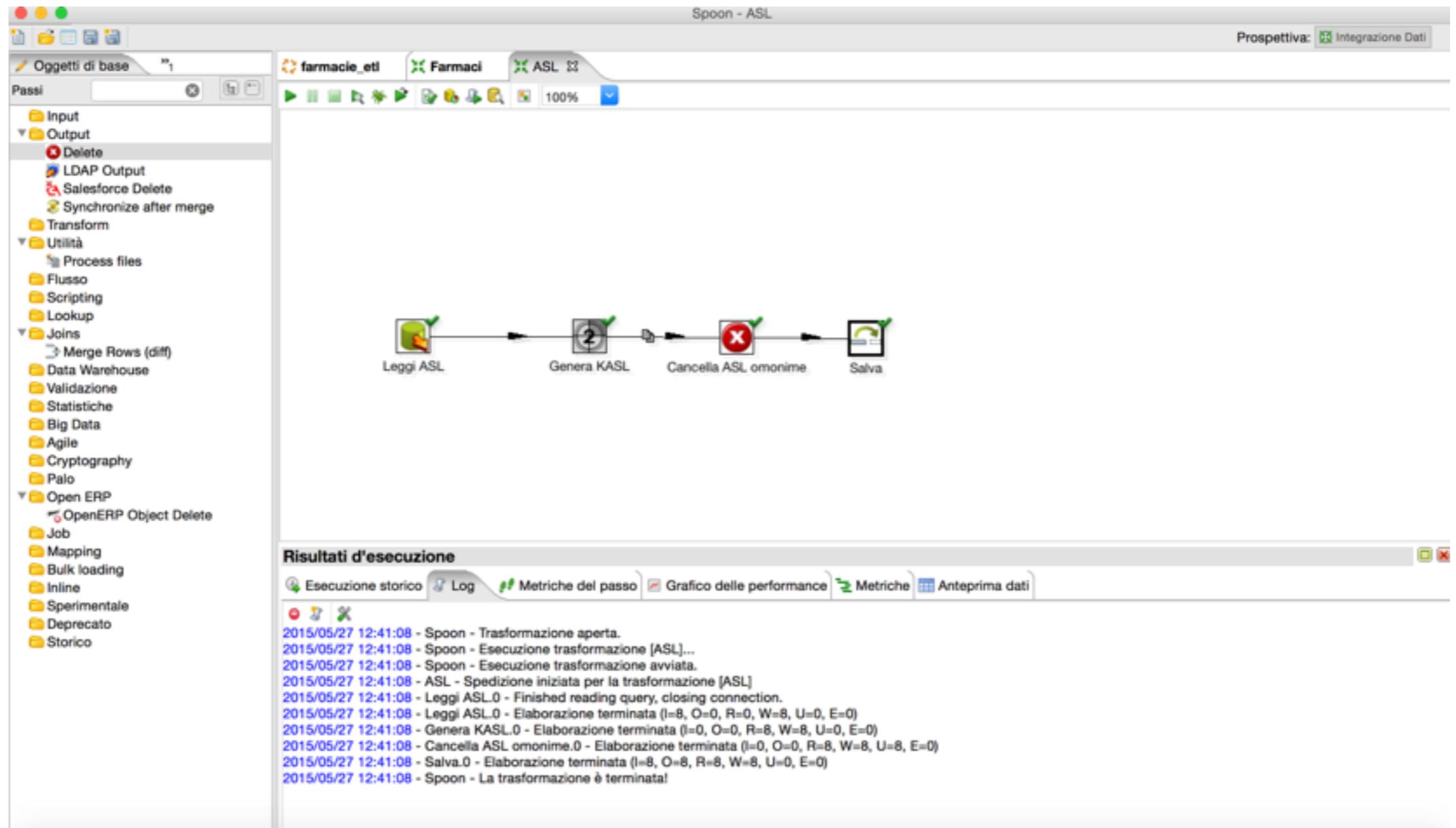
Test trasformazione



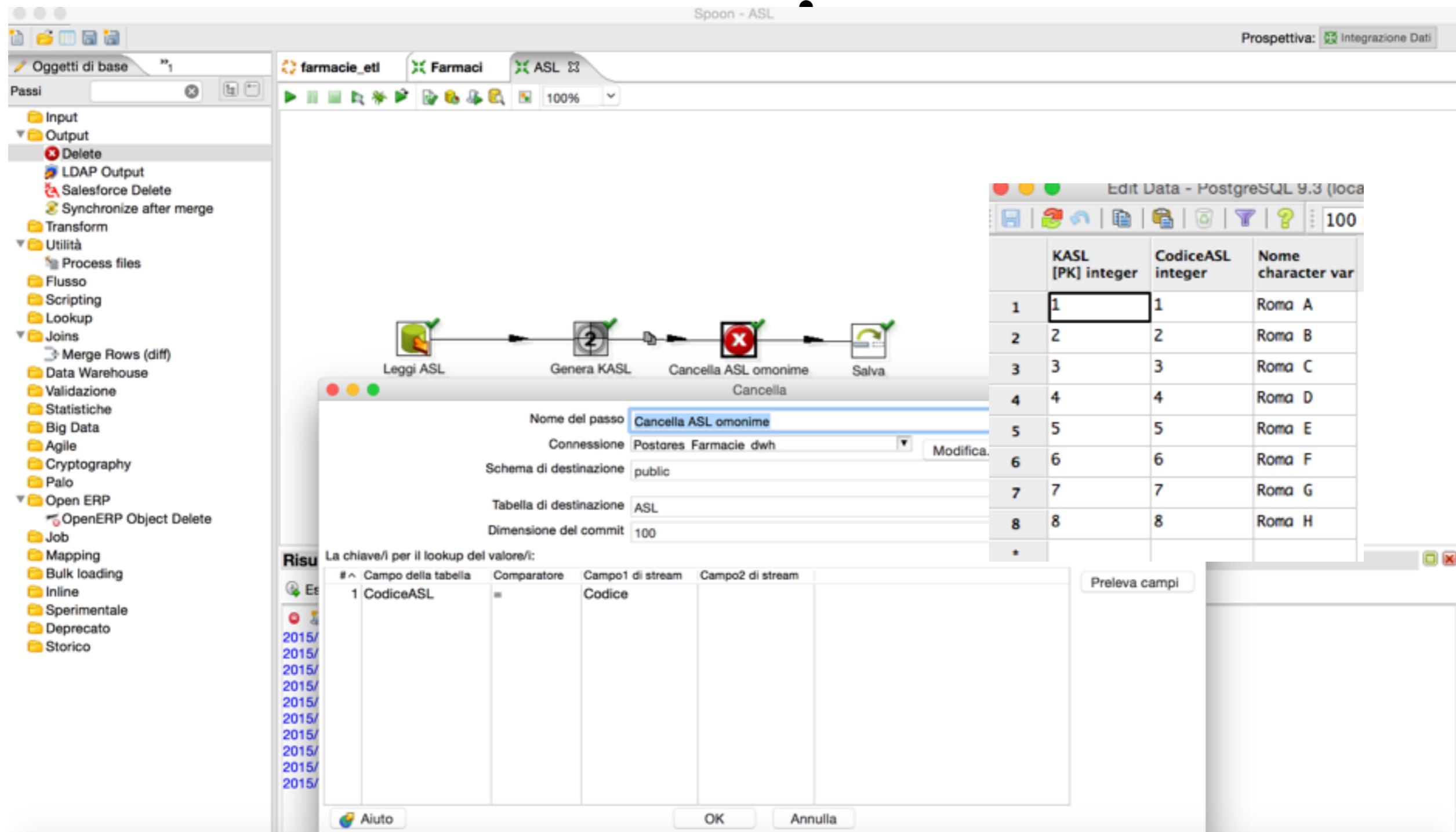
Condividere Data Source



Dimensione ASL



Cancellazione ASL



Dimensioni che non necessitano ETL

- Fasce di età
- Date
- La costruzione si fa in maniera statica:
 - Script PSQL.
 - Programma Java.
 - Flusso di ETL senza sorgenti.

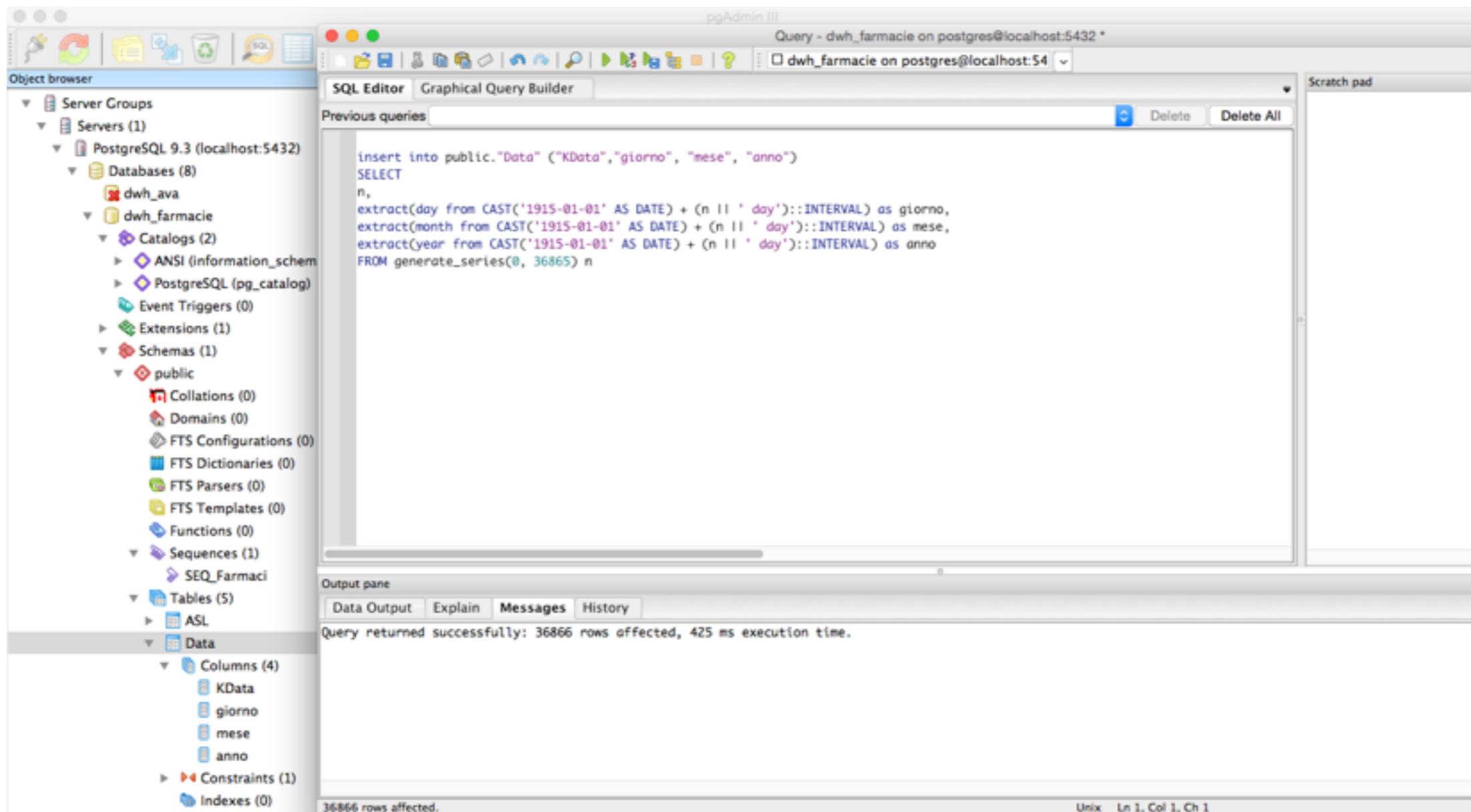
Fasce di età

The screenshot shows the Navicat Premium interface. On the left, the object browser displays a tree structure of databases and tables. In the center, a grid view window is open, showing the contents of the 'FasciaEta' table from the 'dwh_farmacie.public' schema. The table has three columns: 'KFasciaEta', 'DescrizioneFascia', and 'da'. The data is as follows:

KFasciaEta	DescrizioneFascia	da
1	0-3	0
2	4-17	4
3	18-40	18
4	41-60	41
5	61-80	61
6	81-100	81

At the bottom of the grid view, there is a SQL command: `update "public"."FasciaEta" set "da"='81', "a"='100' where "KFasciaEta"=`. The status bar at the bottom indicates "1 of 5 selected".

Data



FattiPrescrizioni

- Una trasformazione per costruire la fact table.
- Si progetta analizzando lo schema della fact table.
- **FattiPrescrizioni(KData, KFarmaco, KRicetta, KASLfarmacia, KASLpaziente, KFasciaEta, Quantita, Importo)**
- Si considerano prima le dimensioni e si valutano quali lookup sono necessari per ottenere le rispettive chiavi surrogate.
- Si valutano le misure e si individua l'origine dei valori.
- Sorgenti per:
 - Ricette join ElementiRicetta (sembra conveniente eseguire il join nel DB).
 - DWH per lookup delle dimensioni.
- Ottengo KData, KFarmaco, KRicetta per codifica (lookup) delle chiavi business in chiavi surrogate nelle dimensioni (lookup nelle dimensioni Data, Farmaci).
 - Ricetta è degenere (si può usare la chiave business).
 - Le altre dimensioni necessitano più attenzione.

FattiPrescrizioni

- KASLfarmacia:
 - Lookup per Farmacie
 - Lookup per Territorio (OLTP)
 - Lookup nella dimensione ASL
- KASLpaziente:
 - Lookup stream per Pazienti
 - Lookup per Territorio (OLTP)
 - Lookup nella dimensione ASL
- KFasciaEta:
 - Lookup stream per Pazienti
 - Calcolo età dalla data di nascita
 - Lookup per range in FasciaEta

Misure

- Quantità
 - Direttamente derivabile da ElementiRicetta
- Importo
 - Lookup Farmaco.
 - Prodotto con quantità.

Input

Spoon - Lookup Farmaci (cambiato)

Prospettiva: Integrazione Dati

Oggetti di base farmacie_etl Farmaci ASL Prescrizioni

Passi

- Input
- Output
- Transform
- Utilità
- Flusso
- Scripting
- Lookup
 - Database lookup
 - Fuzzy match
 - Stream lookup
- Joins
- Data Warehouse
 - Combination
 - Dimension lookup
- Validazione
- Statistiche
- Big Data
- Agile
- Cryptography
- Palo
- Open ERP
- Job
- Mapping
- Bulk loading
- Inline
- Sperimentale
- Deprecato
- Storico
 - Stream lookup

Nome passo: Leggi Ricette e ElementiRicette
Connessione: Postgres_Farmacie

SQL:

```
select
    R."Numero",
    R."CodFarmacia",
    R."CFPaziente",
    extract(day from "Data") as giorno,
    extract(month from "Data") as mese,
    extract(year from "Data") as anno,
    E."NumeroRicetta",
    E."CodFarmaco",
    E."Quantita"
  from
    public."Ricette" R join public."ElementiRicetta" E
  on (R."Numero" = E."NumeroRicetta")
```

Tabella d'input

Righe di passo: Leggi Ricette e ElementiRicetta (21 righe)

#^	Numero	CodFarmacia CFPaziente	giorno	mese	anno	NumeroRicetta	CodFarmaco	Quantita
1	1	1 AMNG77M42L418L	2	5	2015	1	1	1
2	1	1 AMNG77M42L418L	2	5	2015	1	2	3
3	2	1 BLADNL71T58D969X	4	4	2015	2	4	5
4	2	1 BLADNL71T58D969X	4	4	2015	2	1	1
5	2	1 BLADNL71T58D969X	4	4	2015	2	3	2
6	3	3 BNTGBR80H25G674I	8	5	2015	3	4	1
7	3	3 BNTGBR80H25G674I	8	5	2015	3	5	5
8	4	4 DCCFNC63R23L219K	6	2	2015	4	5	2
9	4	4 DCCFNC63R23L219K	6	2	2015	4	19	1
10	4	4 DCCFNC63R23L219K	6	2	2015	4	7	1
11	5	5 GRDPLA57E30D742Q	13	3	2015	5	6	1

Linea 7 Colonna 7

Abilita la conversione lazy

Sostituisci le variabili nello script?

Inserire i dati dal passo

Eseguire per ogni riga?

Dimensione limite: 0

Alto OK Chiudi Mostra log

Lookup Farmaco

Spoon - Lookup Farmaci

Prospettiva: **Integrazione Dati**

Oggetti di base

Passi

- Input
 - HL7 Input
 - Json Input
 - Yaml Input
- Output
 - Automatic Documentation Output
 - Json output
 - LDAP Output
 - Microsoft Access Output
 - Microsoft Excel Output
 - Pentaho Reporting Output
 - Properties Output
 - RSS Output
 - S3 File Output
 - SQL File Output
 - Table output
 - Text file output
 - XML Output
- Transform
 - Concat Fields
 - Row denormaliser
- Utilità
 - Delay row
- Flusso
 - Detect empty stream
 - Scripting
- Lookup
 - Fuzzy match
- Joins
 - Join Rows (cartesian product)
 - Merge Join
- Data Warehouse
- Validazione
- Statistiche
 - Output steps metrics
- Big Data
 - Cassandra Output
 - Hadoop File Output
 - HBase Output
 - MapReduce Output

Lookup valori di database

Nome del passo: Add prezzo

Connessione: Postgres Farmacie

Schema di lookup: public

Tabella di lookup: "Farmaci"

Abilitare cache?

Dimensione cache in righe (0=tutto nella cache): 0

Caricare tutti i dati dalla tabella?

La chiave/i di lookup per il valore/i:

#	Campo della tabella	Comparatore	Campo1	Campo2
1	Codice	=	CodFarmaco	

I valori da ritornare dalla tabella di lookup:

#	Campo	Nuovo nome	Default	Tipo
1	Prezzo			None

Non passare la riga se il lookup fallisce

Failire sui risultati multipli?

Ordinato per:

Exception:

Età:

Add prezzo

Elimina attributi inutili

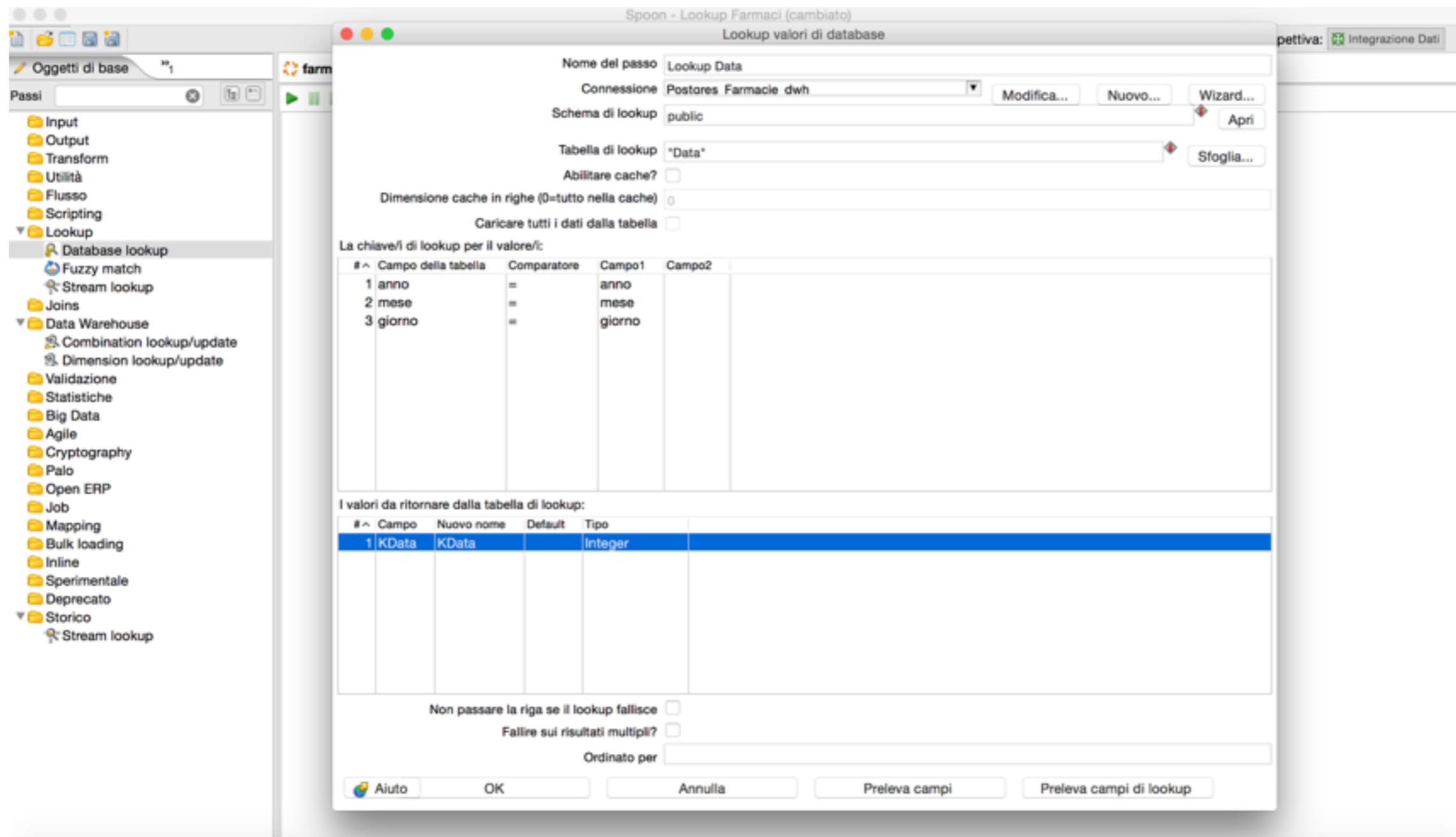
Aggiungi a FactTable

Exception:

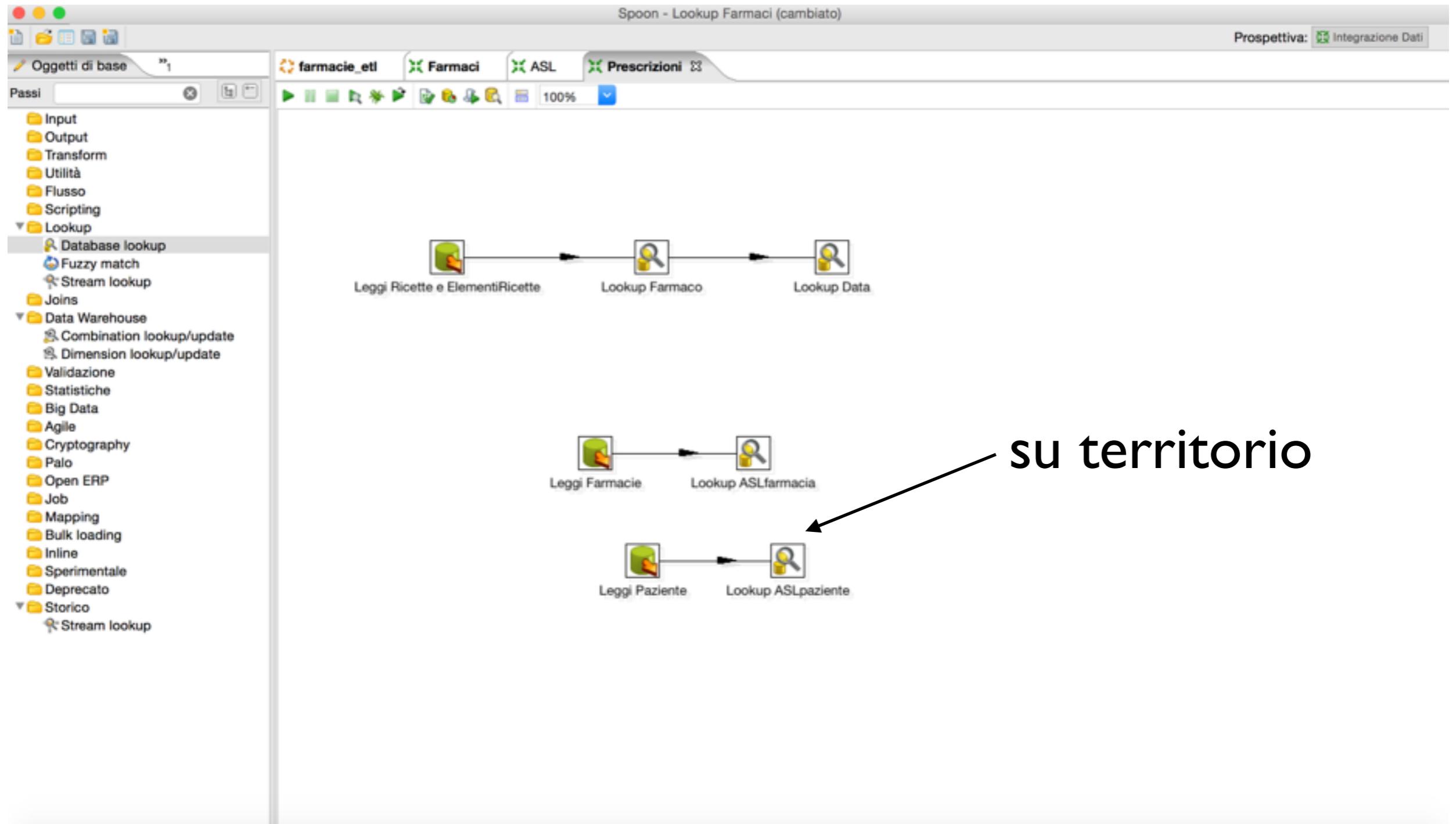
Alto OK Annulla Preleva campi Preleva campi di lookup

2015/05/27 10:41:42 - Add prezzo.v0 - at org.pentaho.di.trans.step.RunThread.run(RunThread.java:62)

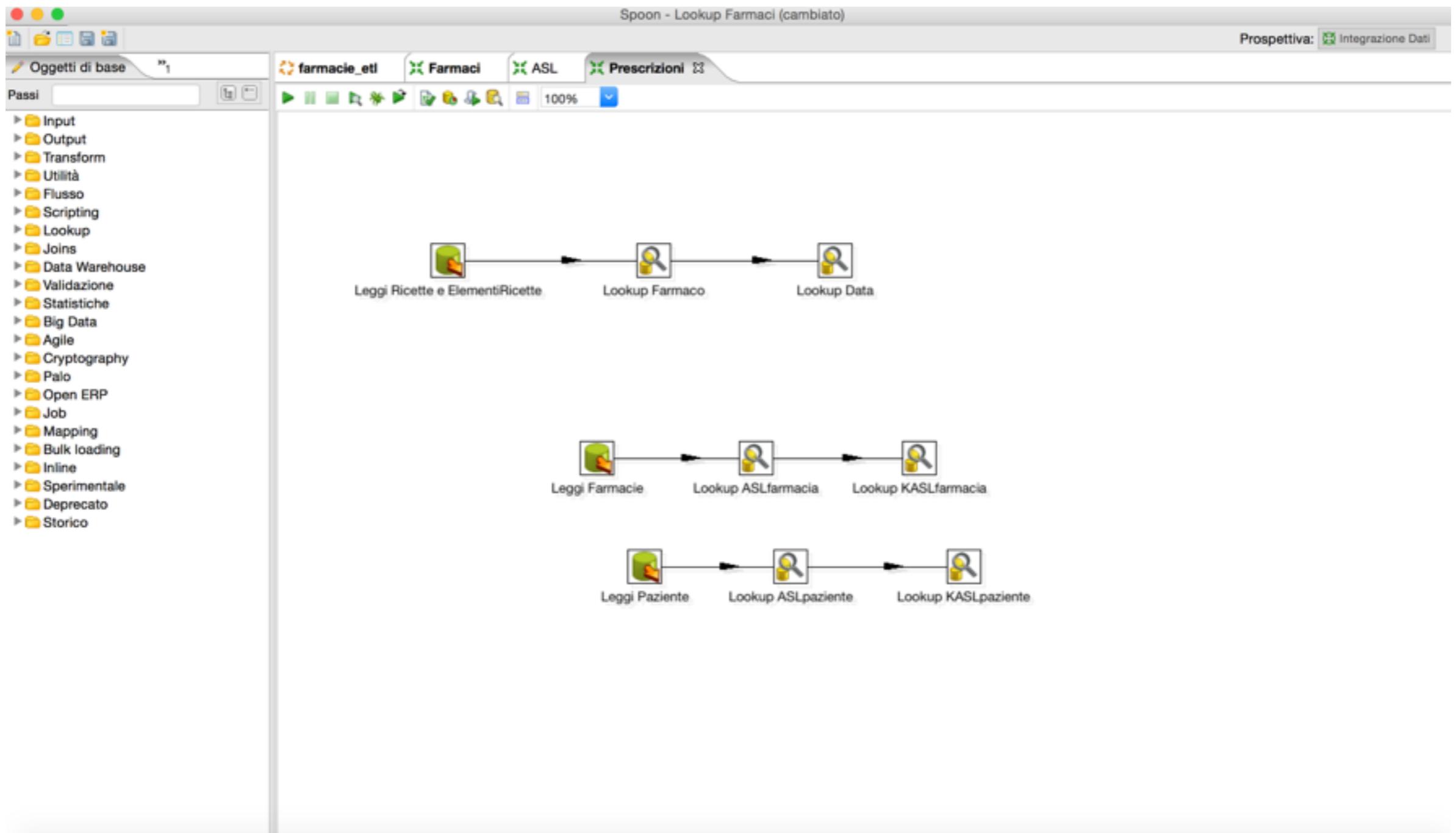
Lookup Data



ASLfarmacia, ASLpaziente



ASLfarmacia, ASLpaziente



KASLpaziente

Spoon - Lookup Farmaci

Prospectiva: Integrazione Dati

Oggetti di base

Passi

- Input
 - HL7 Input
 - Json Input
 - Yaml Input
- Output
 - Automatic Documentation Output
 - Json output
 - LDAP Output
 - Microsoft Access Output
 - Microsoft Excel Output
 - Pentaho Reporting Output
 - Properties Output
 - RSS Output
 - S3 File Output
 - SQL File Output
 - Table output
 - Text file output
 - XML Output
- Transform
 - Concat Fields
 - Row denormaliser
- Utilità
 - Delay row
- Flusso
 - Detect empty stream
 - Scripting
- Lookup
 - Fuzzy match
- Join
 - Join Rows (cartesian product)
 - Merge Join
- Data Warehouse
- Validazione
- Statistiche
 - Output steps metrics
- Big Data
 - Cassandra Output
 - Hadoop File Output
 - HBase Output
 - MapReduce Output

Lookup valori di database

Nome del passo: Lookup KASLpaziente
Connessione: Postgres_Farmacie_dwh
Schema di lookup: public
Tabella di lookup: ASL
Abilitare cache?
Dimensione cache in righe (0=tutto nella cache)
Caricare tutti i dati dalla tabella

La chiave/i di lookup per il valore/i:

#	Campo della tabella	Comparatore	Campo1	Campo2
1	CodiceASL	=	ASL	

I valori da ritornare dalla tabella di lookup:

#	Campo	Nuovo nome	Default	Tipo
1	KASL	KASLpaziente		Integer

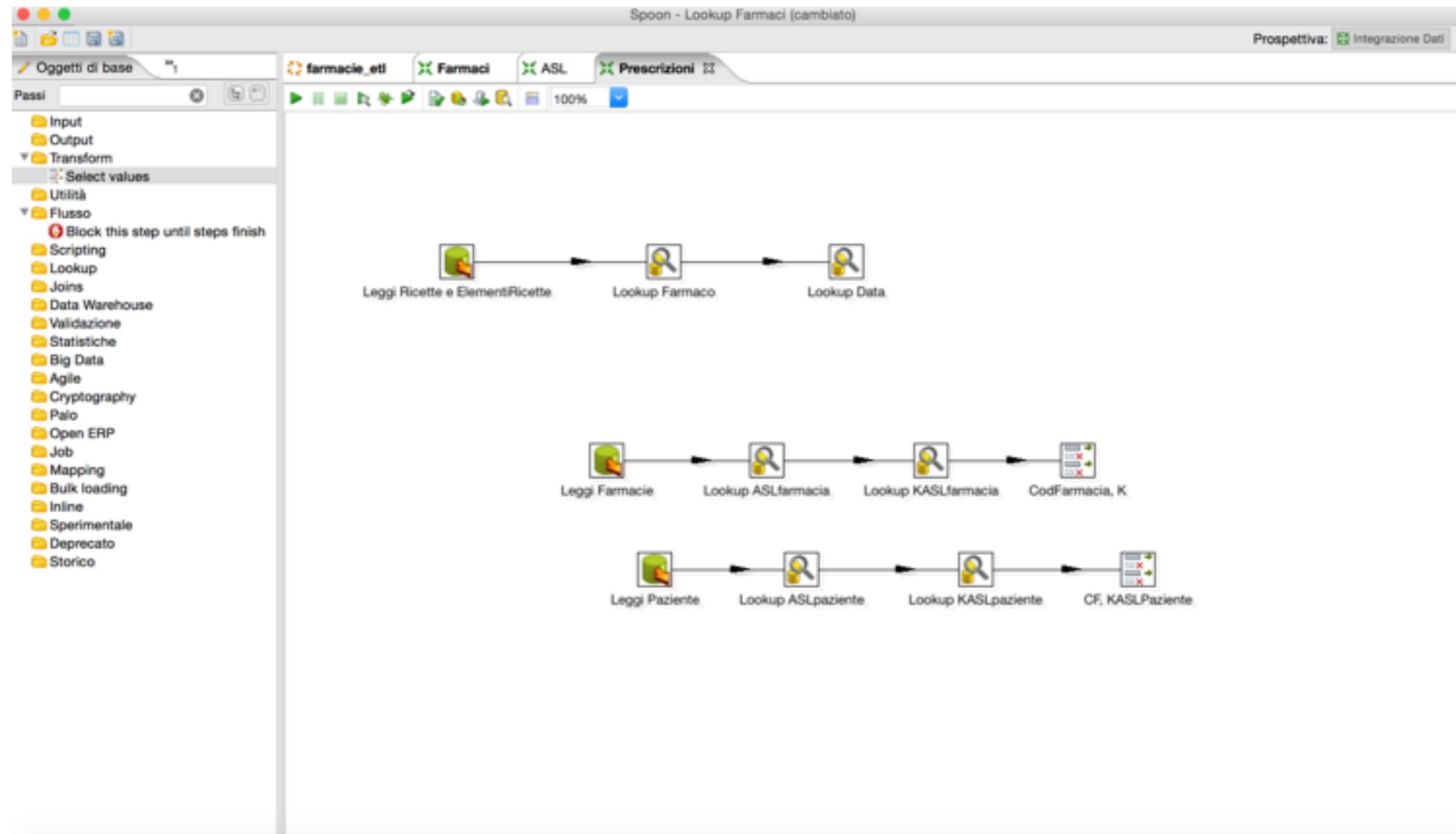
Non passare la riga se il lookup fallisce
Fallire sui risultati multipli?
Ordinato per:

OK Annula Preleva campi Preleva campi di lookup

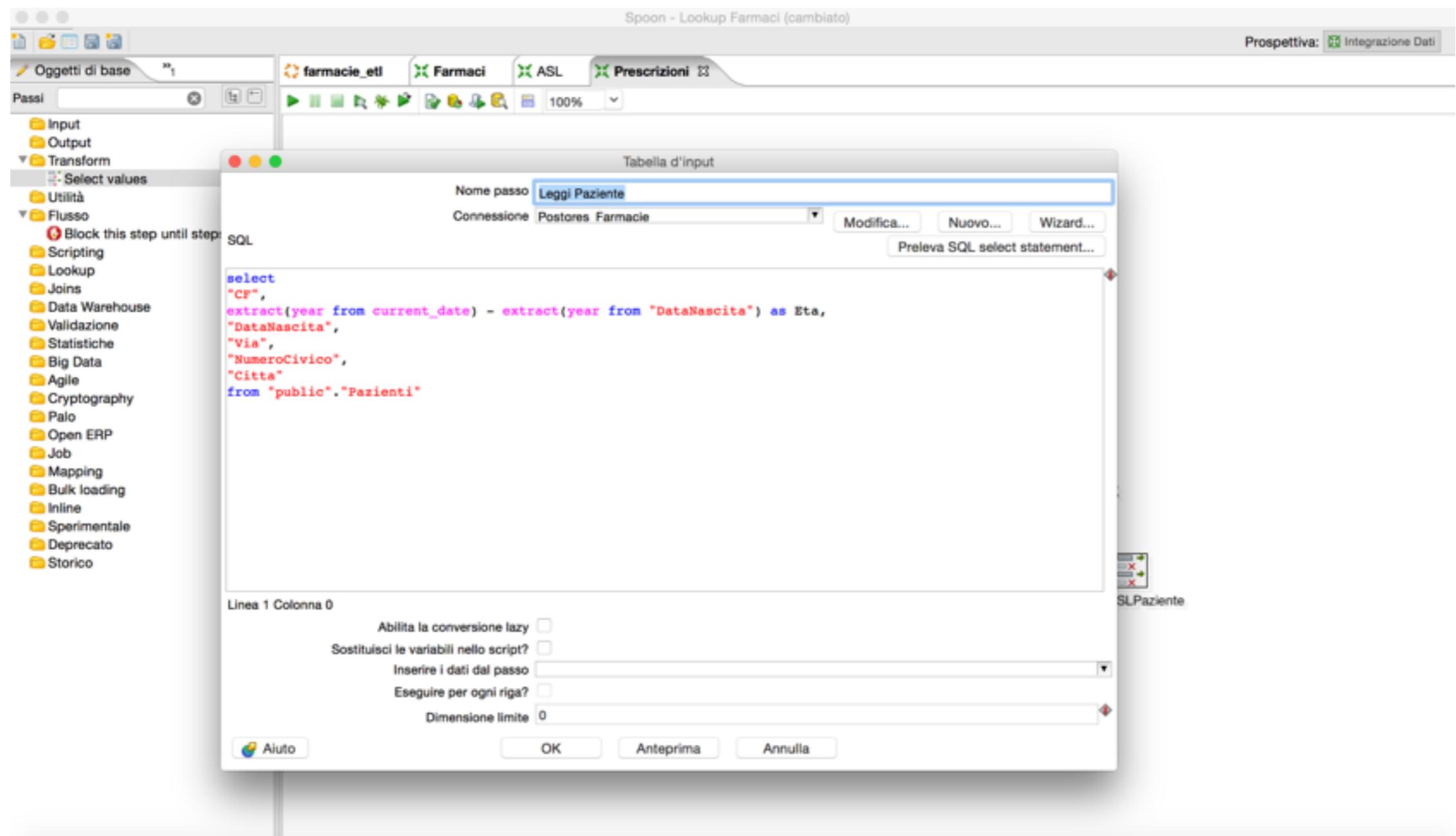


.KettleStepException:

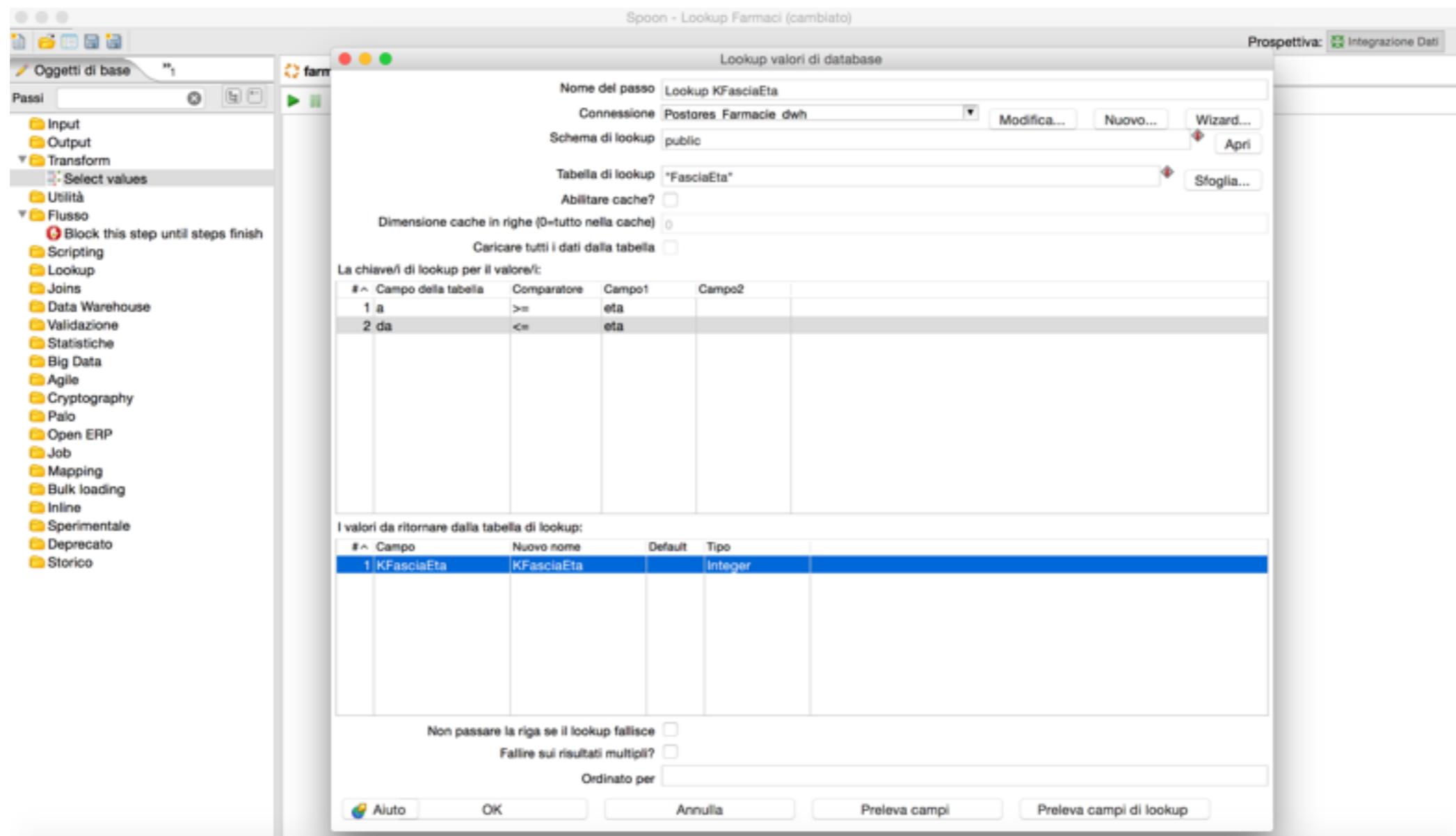
KASLfarmacia, KASLpaziente



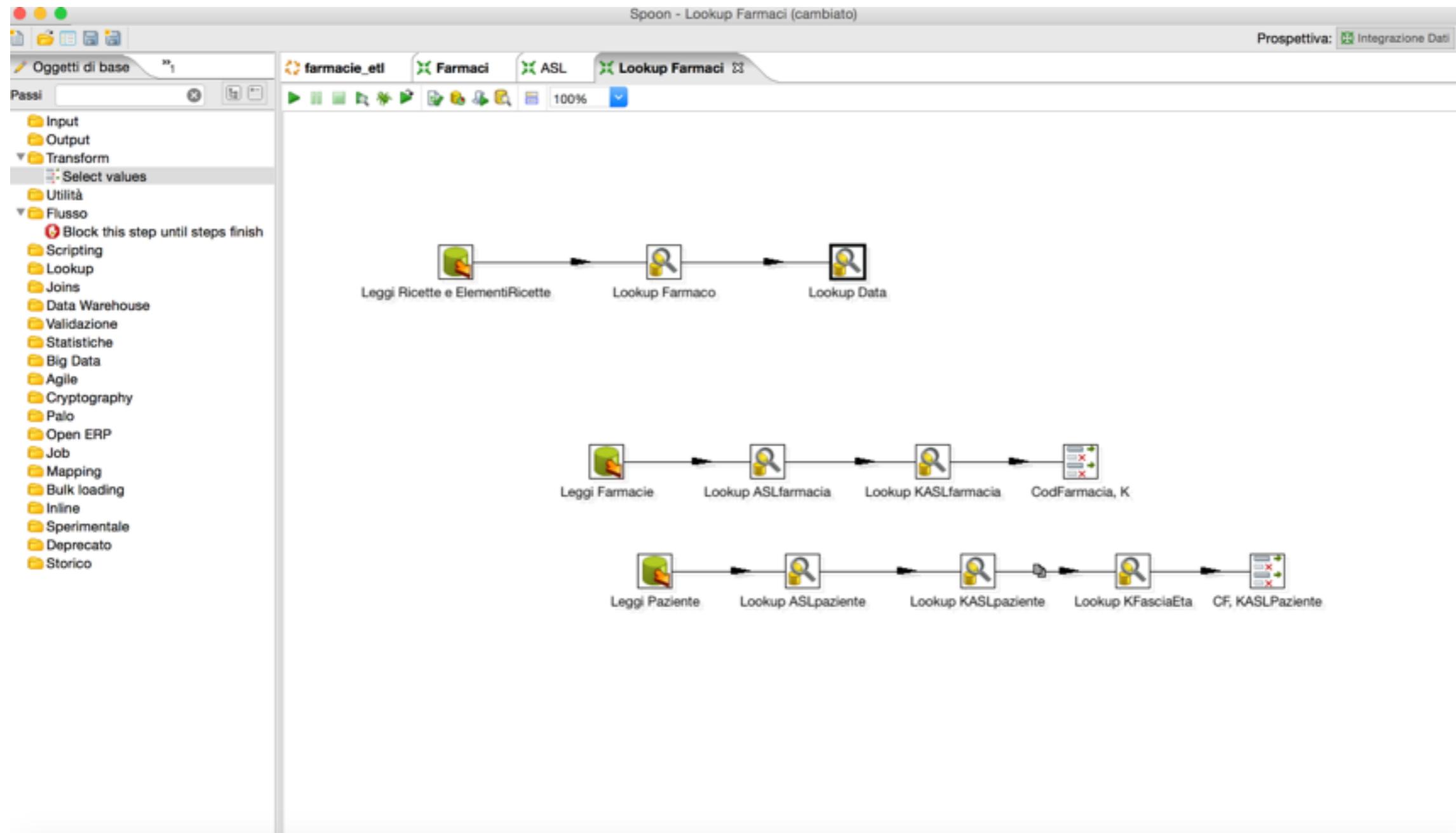
KFasciaEta (input Paziente)



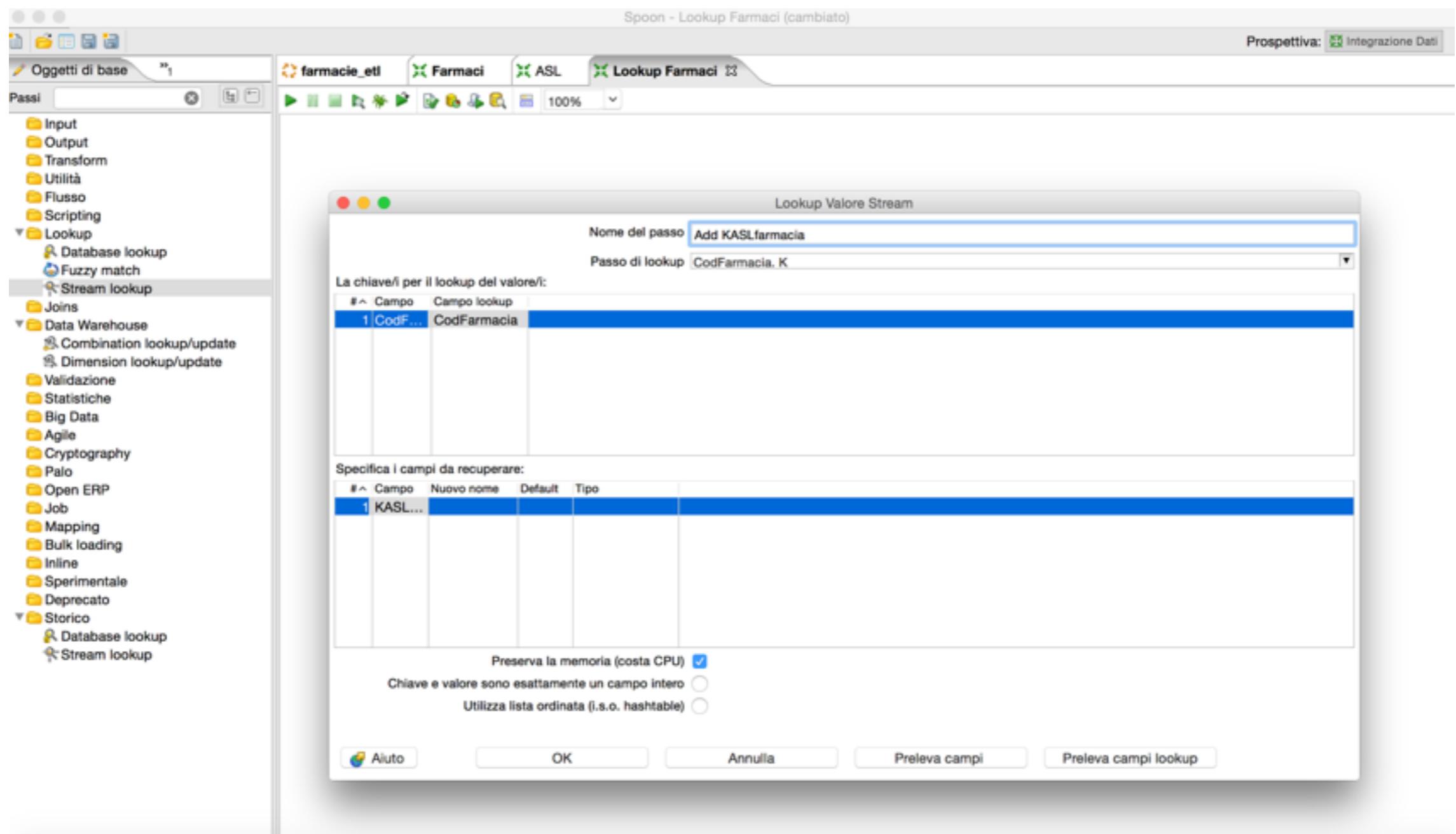
Lookup KFasciaEta



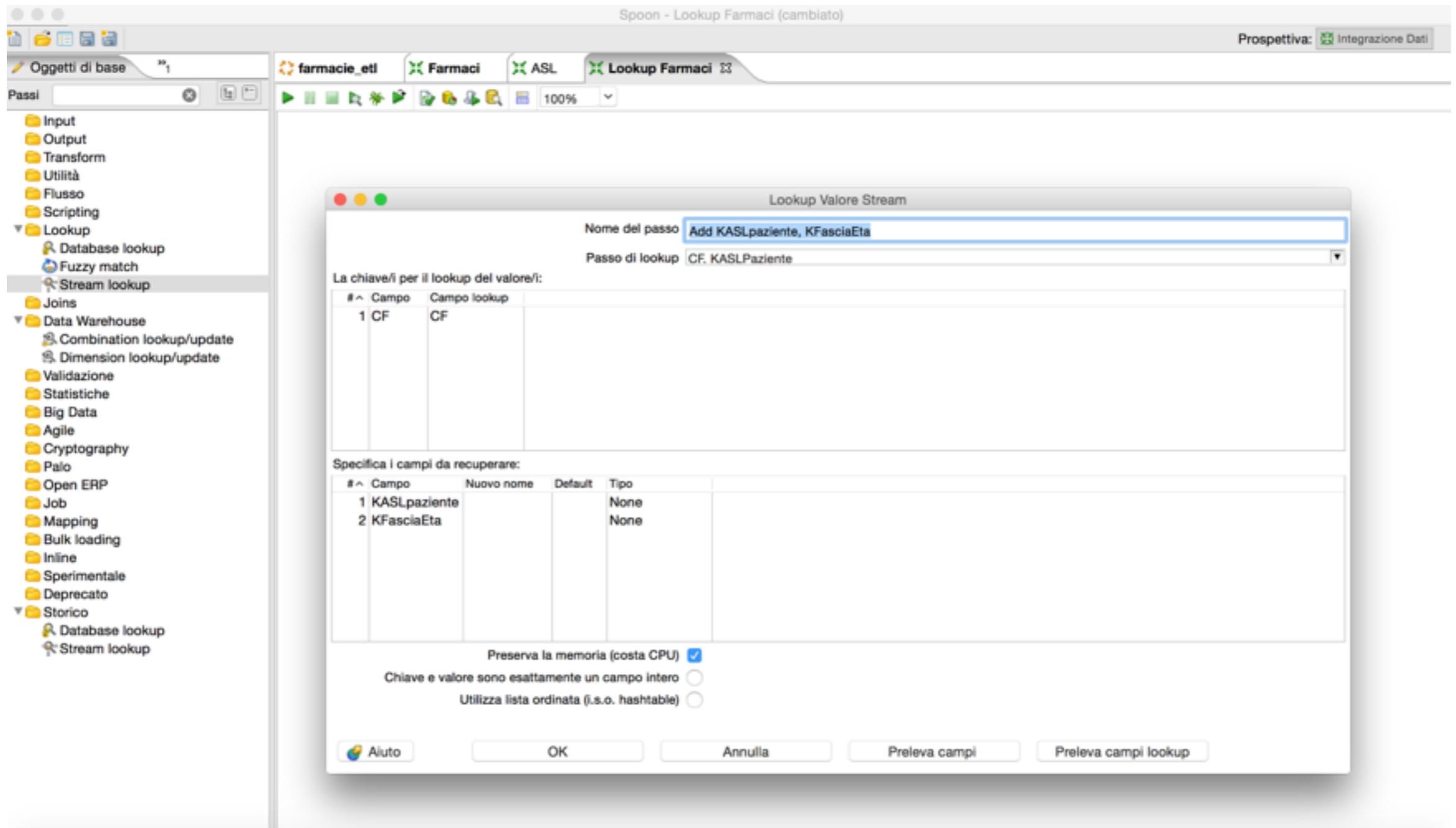
Unire i vari stream



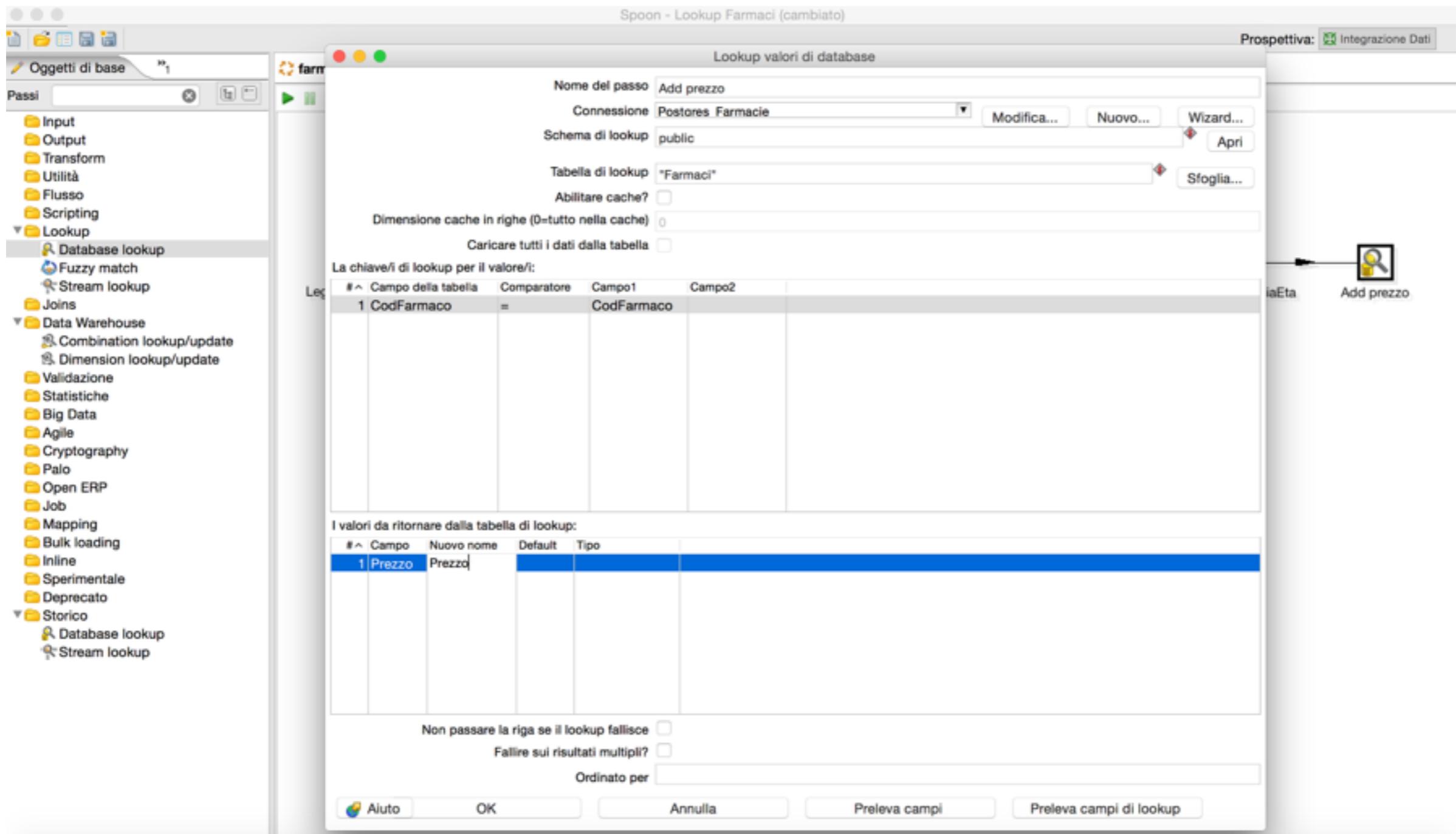
Aggiungere KASLfarmacia allo stream superiore



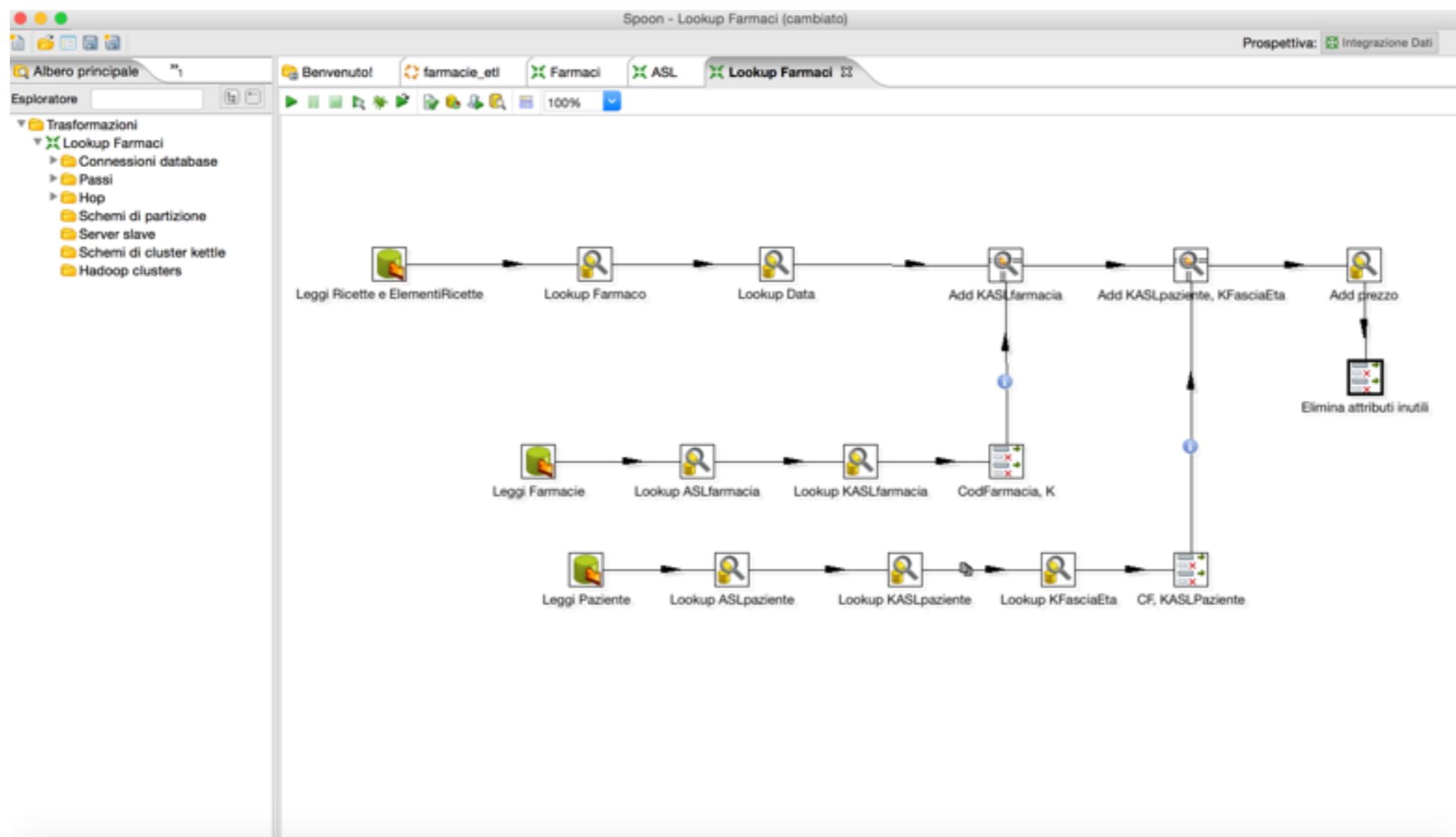
Aggiungere KASLpaziente, KFasciaETA allo stream superiore



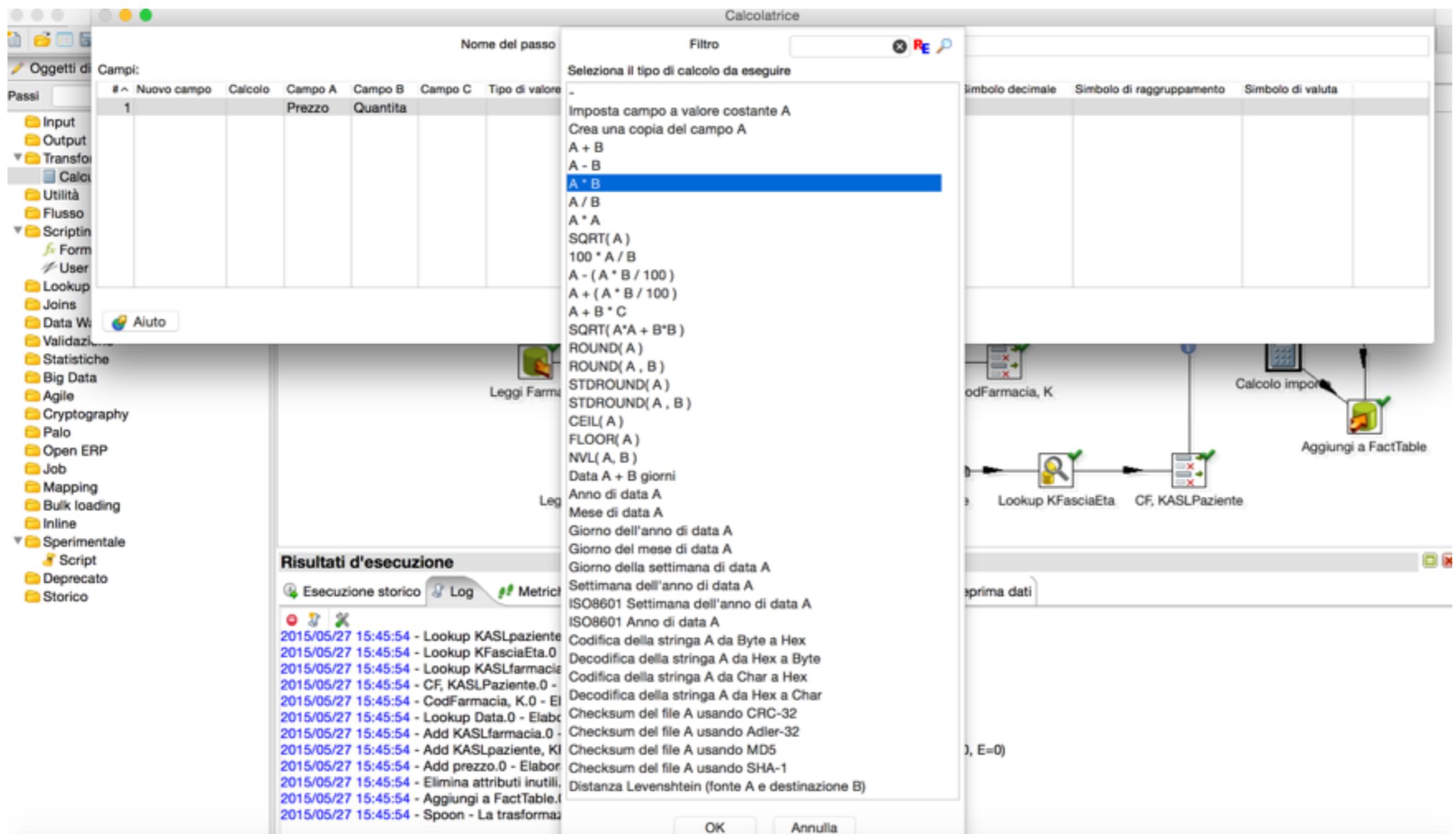
Lookup della misura Prezzo



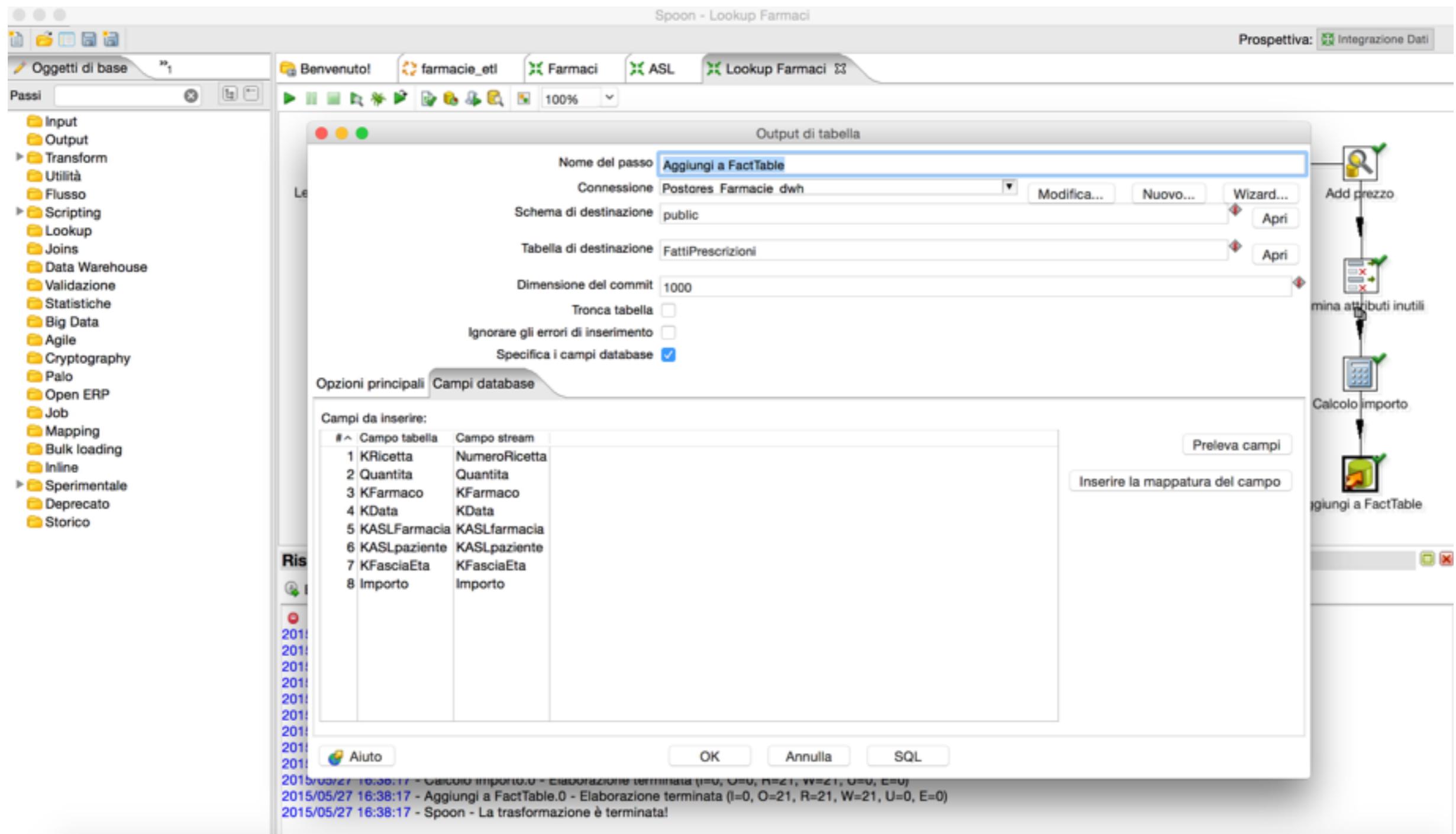
Prepararsi alla scrittura



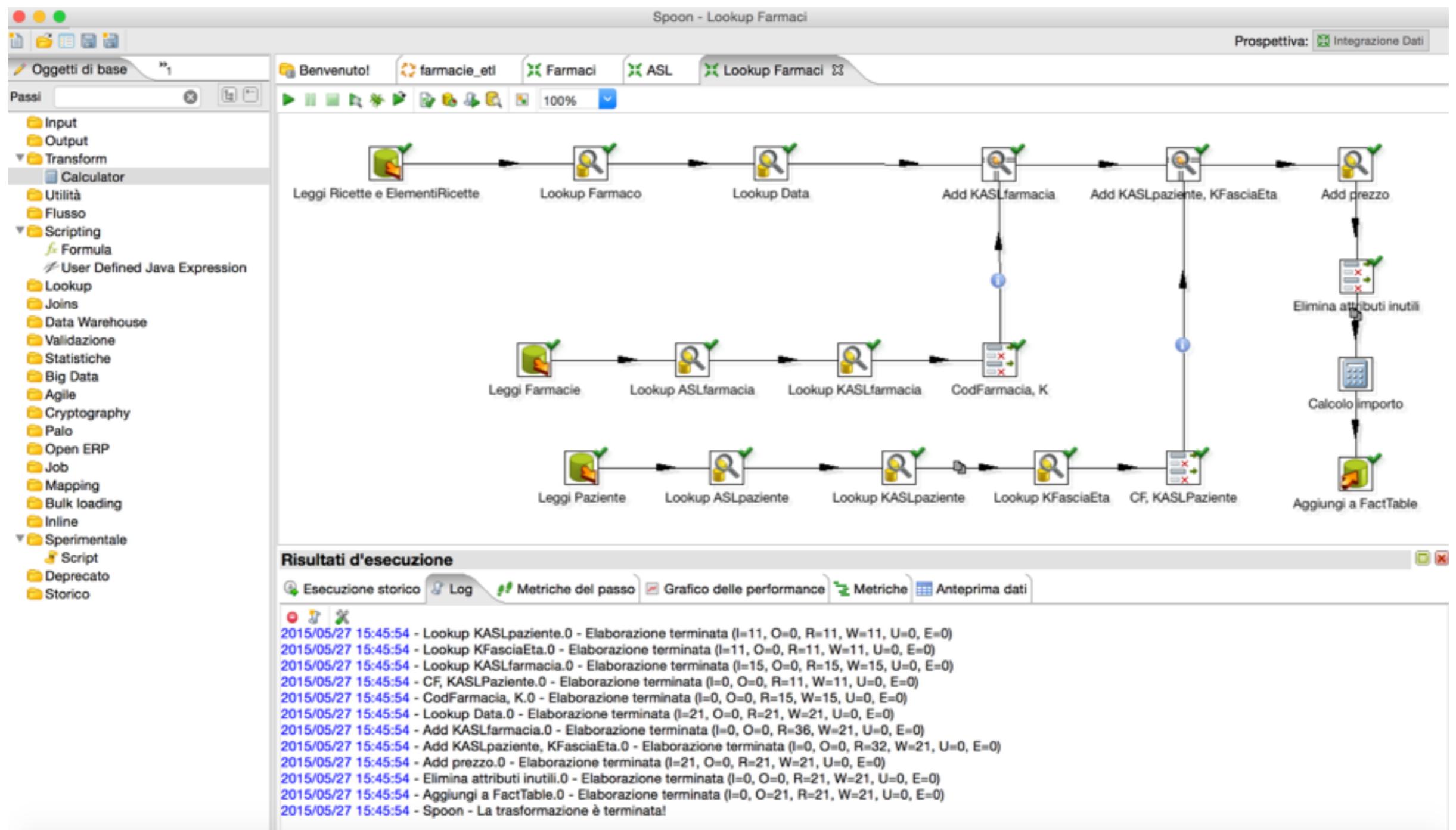
Calcolo Importo



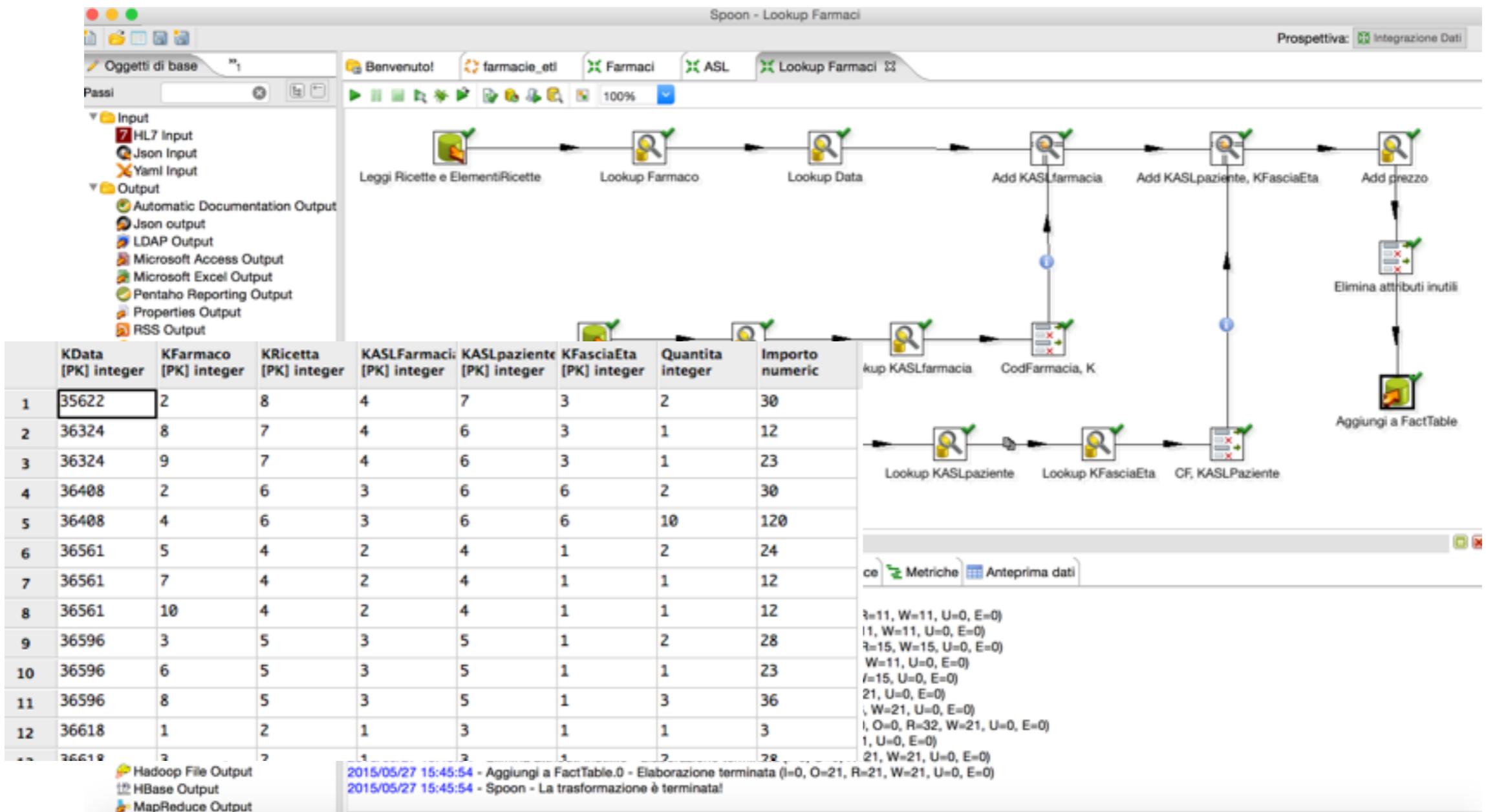
Mapping di output



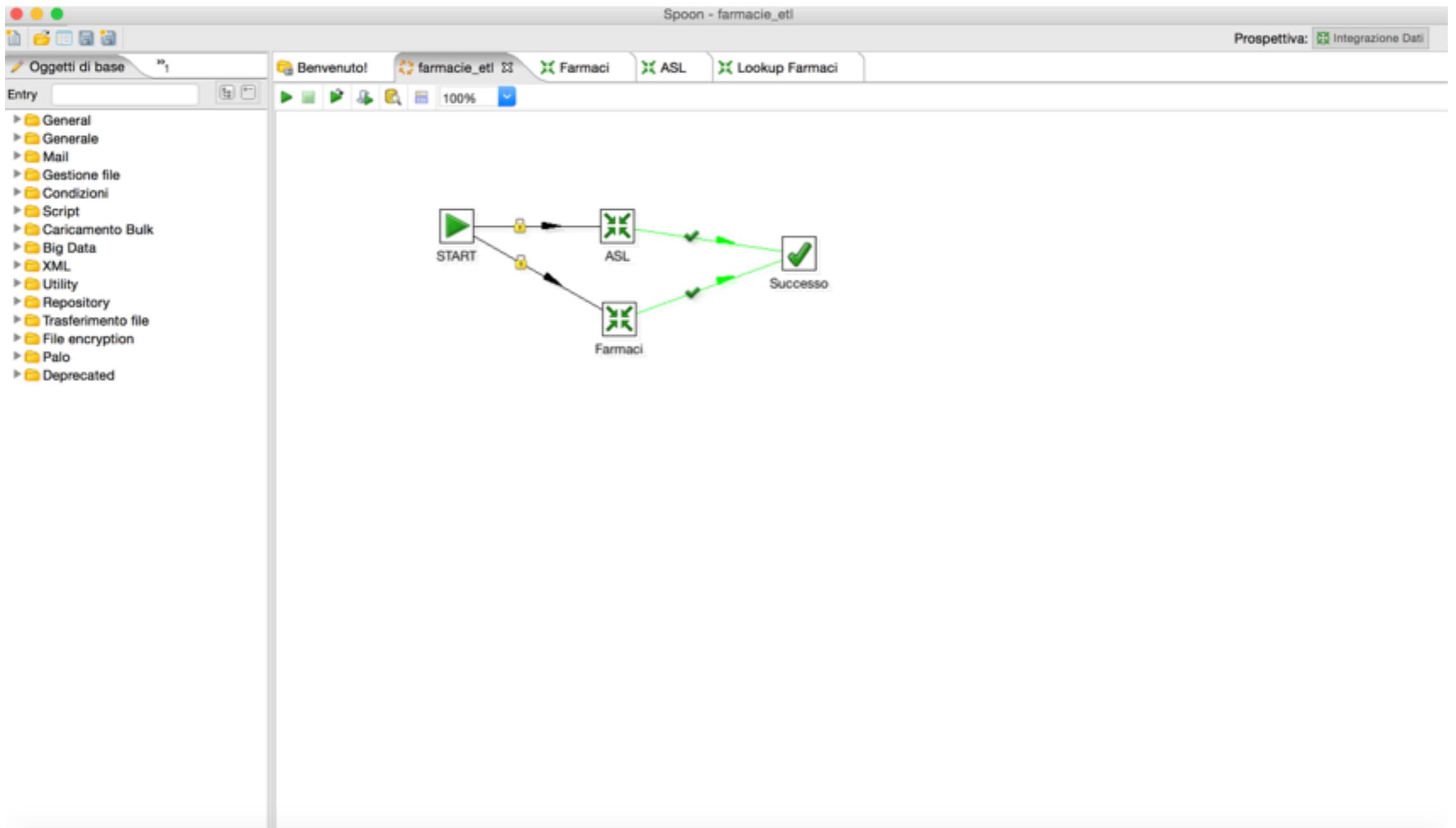
L'intera trasformazione



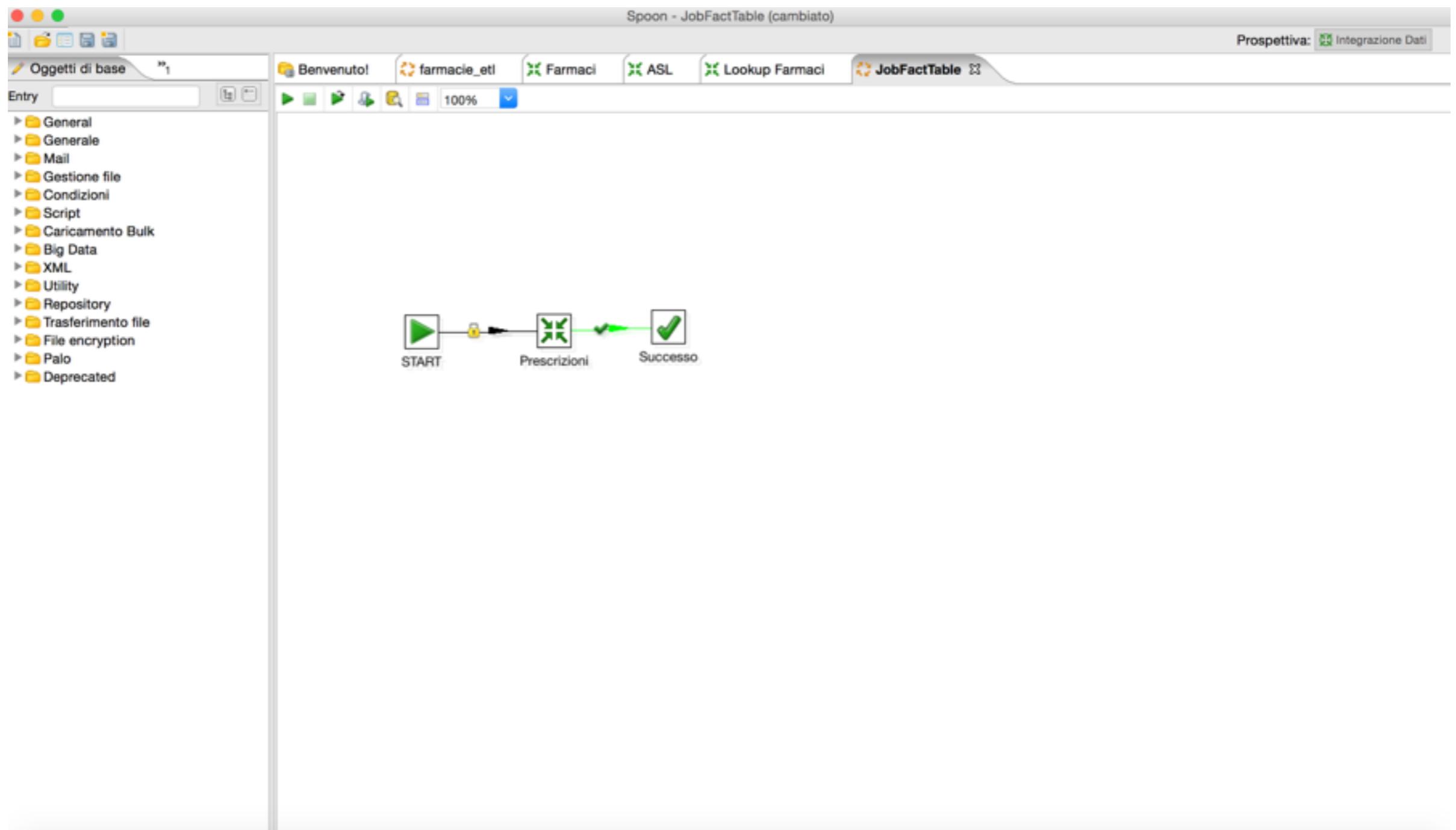
Test della trasformazione



Il Job per le dimensioni



Il Job per la Fact Table



Crediti e riferimenti

- *Matteo Golfarelli, Stefano Rizzi - Data Warehouse - Teoria e pratica della progettazione (McGraw Hill)*
- [https://anonymousbi.wordpress.com/2013/12/15/
pentaho-bi-server-5-0-ice-mysql-installation-guide/](https://anonymousbi.wordpress.com/2013/12/15/pentaho-bi-server-5-0-ice-mysql-installation-guide/)
- <http://infocenter.pentaho.com/help/index.jsp>