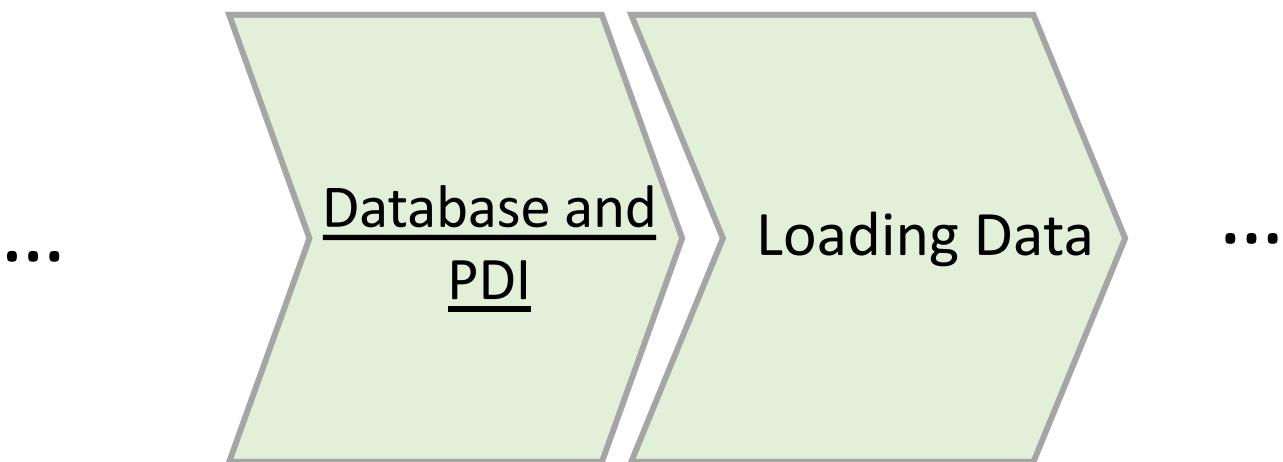


DATA ANALYTICS (Data Warehouse)

Pentaho Data Integration

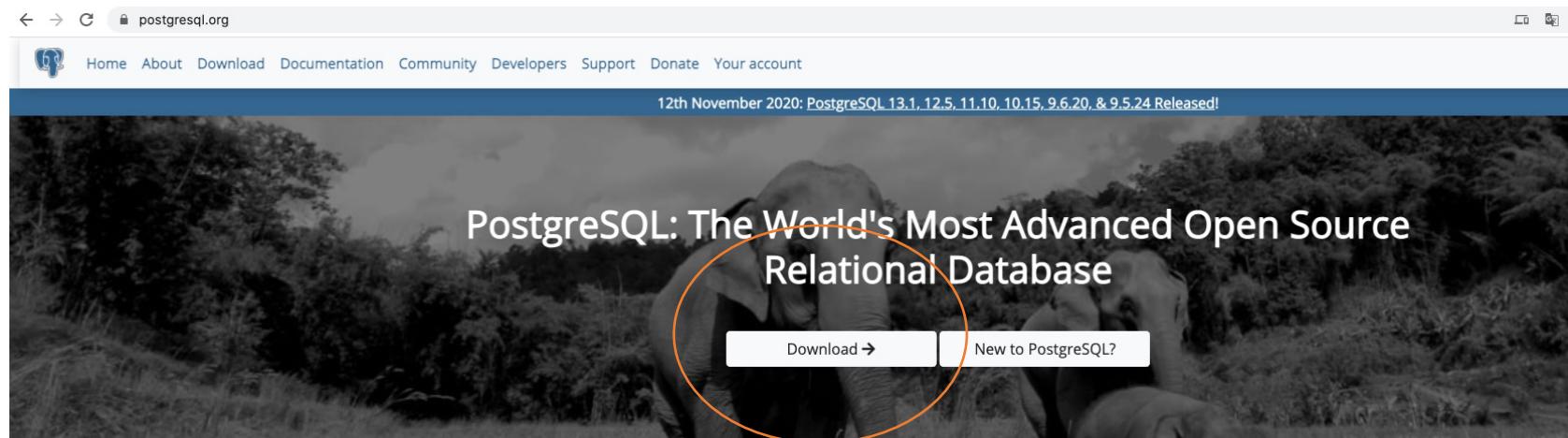
Luca Cinelli, PhD
luca.cinelli@unical.it

Outline



Setup PostgreSQL

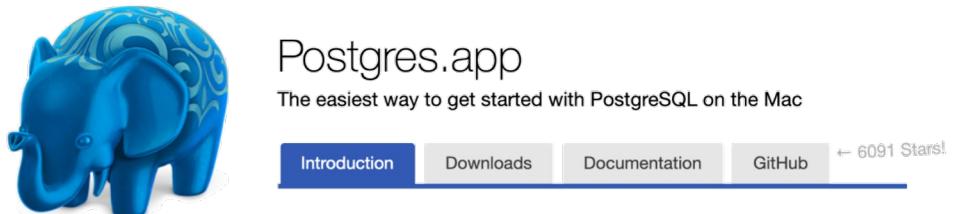
- <https://www.postgresql.org/>



- Database engine
- Graphical Interface: PGAdmin

Setup PostgreSQL

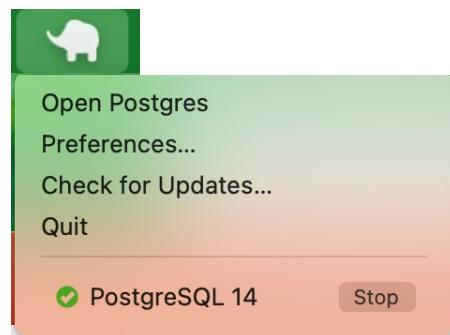
- <https://www.postgresapp.com>



Postgres.app is a full-featured PostgreSQL installation packaged as a standard Mac app. It includes everything you need to get started, and we've even included the popular extension [PostGIS](#) for geo data.

Postgres.app has a beautiful user interface and a convenient menu bar item. You never need to touch the command line to use it – but of course we do include all the necessary [command line tools](#) and header files for advanced users.

Postgres.app can install minor updates automatically, so you get bugfixes as soon as possible.



- Database engine
- Graphical Interface: PGAdmin

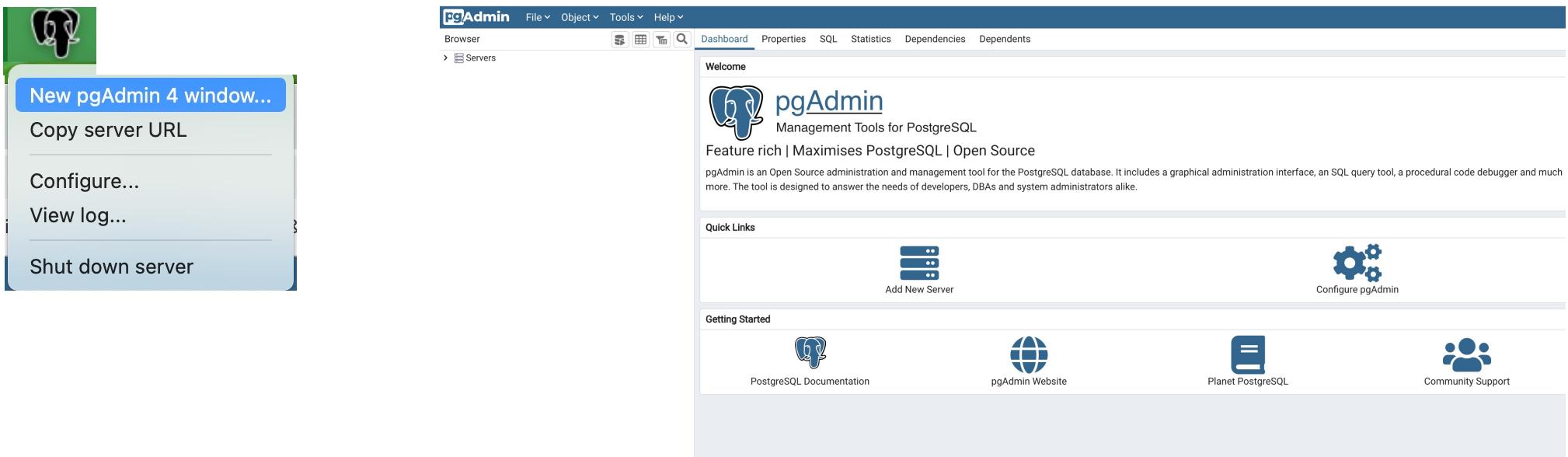
Installing Postgres.app

- ① Download → Move to Applications folder → Double Click
If you don't move Postgres.app to the Applications folder, you will see a warning about an unidentified developer and won't be able to open it.
- ② Click "Initialize" to create a new server
- ③ Configure your \$PATH to use the included command line tools (optional):

```
sudo mkdir -p /etc/path.d &&
echo /Applications/Postgres.app/Contents/Versions/latest/bin_1_sudo_tee /etc/naths.d/nostoresann
```

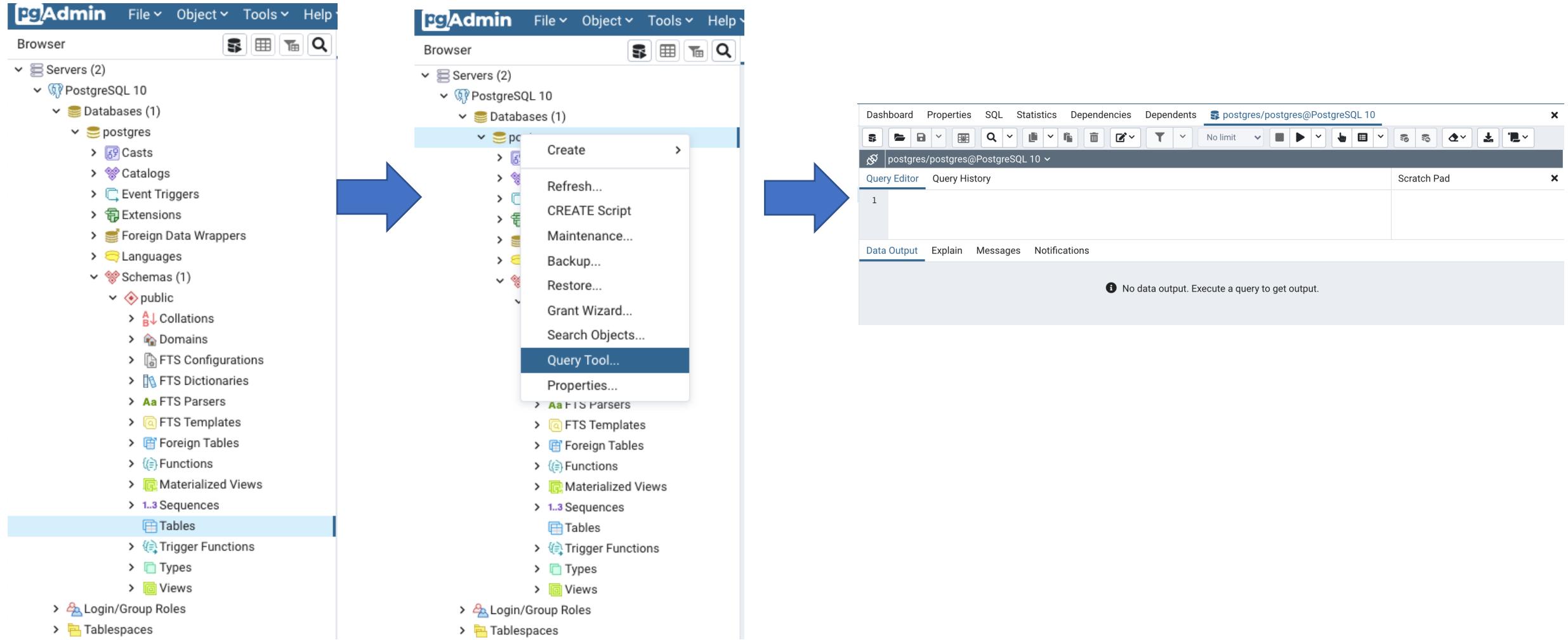
Setup PostgreSQL

- <https://www.postgresapp.com>
- Database engine
- Graphical Interface: PGAdmin (<http://127.0.0.1:51765/browser/>)



Creating tables in PostgreSQL

<http://127.0.0.1:60624/browser/>



Creating tables in PostgreSQL – Sales Table schema example

```
Create table Sales (
    Order_Line int primary key,  

    Order_ID varchar,  

    Order_Date date,  

    Ship_Date date,  

    Ship_Mode varchar,  

    Customer_ID varchar,  

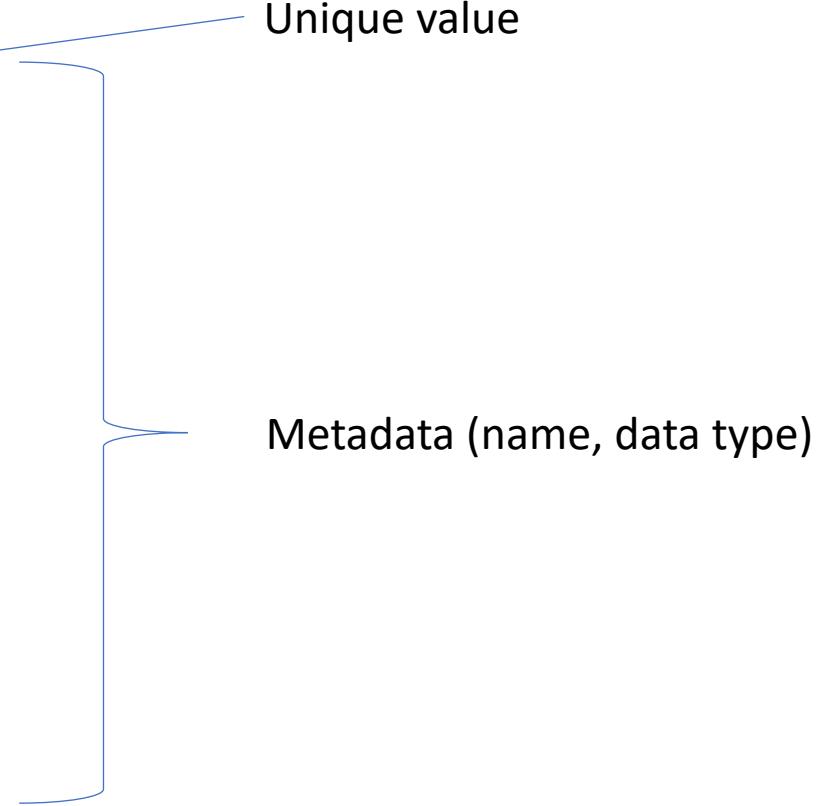
    Product_ID varchar,  

    Sales numeric,  

    Quantity int,  

    Discount numeric,  

    Profit numeric
);
```



Unique value

Metadata (name, data type)

Creating tables in PostgreSQL – Sales Table schema example

The screenshot shows the pgAdmin 4 interface for PostgreSQL version 10. The left sidebar is titled 'Browser' and lists various database objects: Foreign Tables, Functions, Materialized Views, Sequences, Tables (1), sales, and Columns (11). The 'Columns (11)' item is currently selected, highlighted with a blue background. The main panel is titled 'Query Editor' and contains the SQL code for creating the 'Sales' table. The code is as follows:

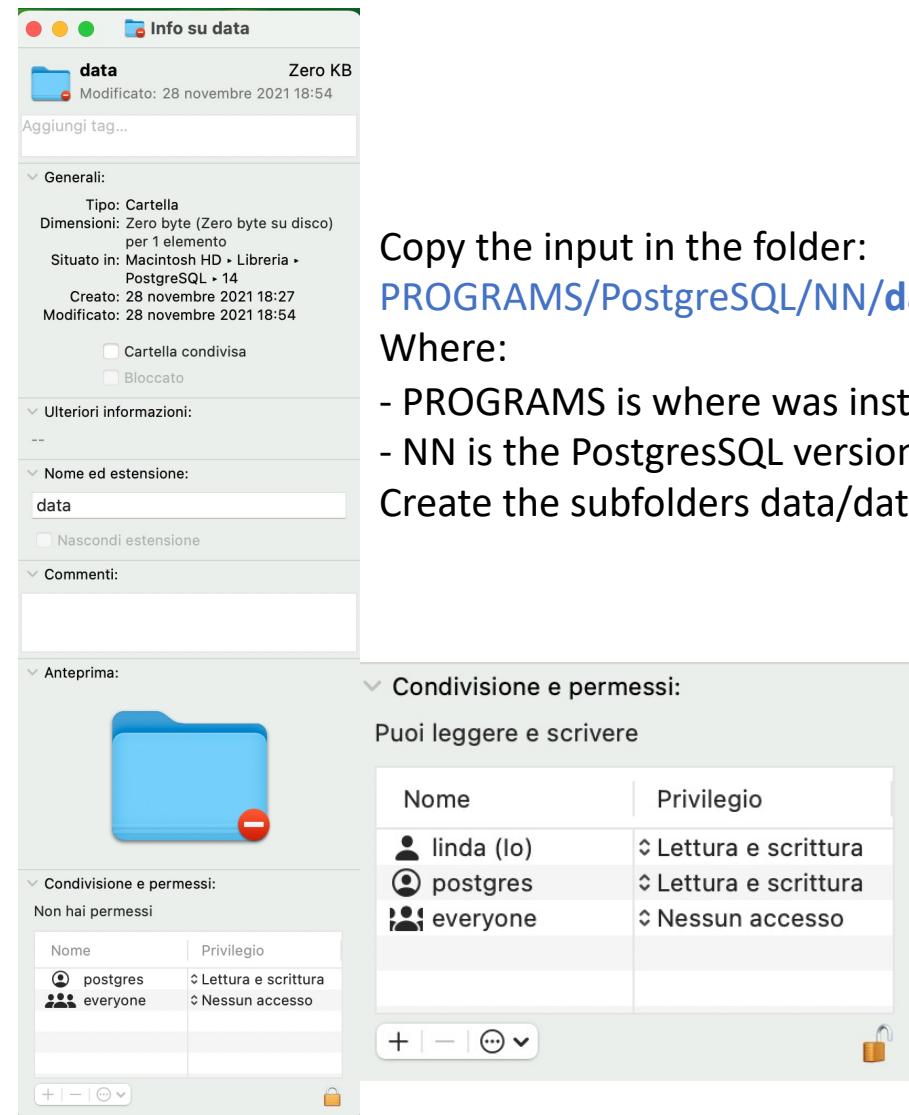
```
1 Create table Sales (
2 Order_Line int primary key,
3 Order_ID varchar,
4 Order_Date date,
5 Ship_Date date,
6 Ship_Mode varchar,
7 Customer_ID varchar,
8 Product_ID varchar,
9 Sales numeric,
10 Quantity int,
```

Below the code, there are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Messages' tab is active, showing the message 'CREATE TABLE'. At the bottom, it says 'Query returned successfully in 61 msec.'

Creating tables in PostgreSQL – Sales Table schema example

Input

SalesforSQL											
Order Line	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Product ID	Sales	Quantity	Discount	Profit	
7973	CA-2017-166142	2020-01-01	2020-01-05	Standard Class	MM-17260	OFF-BI-10004094	26.55	3	0	13.01	
7974	CA-2017-166142	2020-01-01	2020-01-05	Standard Class	MM-17260	FUR-TA-10004607	310.44	3	0.3	-48.78	
7975	CA-2017-164378	2020-01-01	2020-01-04	Second Class	MM-18055	OFF-AR-10001177	6.56	2	0	1.9	
7976	CA-2017-164378	2020-01-01	2020-01-04	Second Class	MM-18055	OFF-LA-10000634	7.83	3	0	3.6	
7977	CA-2017-164378	2020-01-01	2020-01-04	Second Class	MM-18055	TEC-AC-10004708	41.9	2	0	8.8	
7978	CA-2017-164378	2020-01-01	2020-01-04	Second Class	MM-18055	FUR-CH-10002084	664.15	6	0.1	88.55	
7979	CA-2017-164378	2020-01-01	2020-01-04	Second Class	MM-18055	OFF-PA-10004519	8.96	2	0	4.39	
7980	US-2017-128951	2020-01-01	2020-01-03	First Class	RS-19420	OFF-AP-10002191	179.94	3	0	50.38	
7981	US-2017-128951	2020-01-01	2020-01-03	First Class	RS-19420	FUR-TA-10004575	872.94	3	0	157.13	
7982	US-2017-128951	2020-01-01	2020-01-03	First Class	RS-19420	OFF-PA-10003177	12.96	2	0	6.22	
7983	US-2017-156909	2020-01-02	2020-01-04	Second Class	SF-20065	FUR-CH-10002774	71.37	2	0.3	-1.02	
7984	CA-2017-109778	2020-01-02	2020-01-07	Standard Class	VM-21685	OFF-AR-10003759	2.91	2	0.2	0.91	
7985	US-2017-152842	2020-01-02	2020-01-09	Standard Class	NF-18385	FUR-CH-10004218	242.35	3	0.2	15.15	
7986	CA-2017-139948	2020-01-03	2020-01-08	Standard Class	SW-20455	FUR-FU-10002597	7.9	2	0.2	2.17	
7987	US-2017-105046	2020-01-03	2020-01-09	Standard Class	BE-11335	TEC-PH-10004536	269.98	2	0	67.5	
7988	US-2017-105046	2020-01-03	2020-01-09	Standard Class	BE-11335	OFF-PA-10004535	99.9	5	0	47.95	
7989	US-2017-105046	2020-01-03	2020-01-09	Standard Class	BE-11335	FUR-FU-10004848	39.08	4	0	14.46	
7990	CA-2017-126662	2020-01-03	2020-01-07	Standard Class	AB-10255	TEC-CO-10004202	479.98	2	0.2	90	
7991	CA-2017-142342	2020-01-03	2020-01-05	Second Class	AJ-10795	OFF-PA-10004609	32.4	5	0	15.55	
7992	CA-2017-142342	2020-01-03	2020-01-05	Second Class	AJ-10795	OFF-EN-10002592	57.9	5	0	28.95	
7993	CA-2017-142342	2020-01-03	2020-01-05	Second Class	AJ-10795	OFF-ST-10002957	10.56	2	0	0	
7994	CA-2017-142342	2020-01-03	2020-01-05	Second Class	AJ-10795	FUR-BO-10002613	1194.17	5	0.15	210.74	
7995	US-2017-142573	2020-01-04	2020-01-09	Standard Class	ML-17410	FUR-TA-10001932	801.6	5	0.5	-448.9	
7996	US-2017-142573	2020-01-04	2020-01-09	Standard Class	ML-17410	FUR-CH-10004218	161.57	2	0.2	10.1	
7997	US-2017-142573	2020-01-04	2020-01-09	Standard Class	ML-17410	OFF-PA-10000246	16.1	2	0.2	5.23	
7998	US-2017-142573	2020-01-04	2020-01-09	Standard Class	MI-17410	OFF-BI-10003350	7.66	4	0.7	-6.12	



Copy the input in the folder:

[PROGRAMS/PostgreSQL/NN/data/dataset](#)

Where:

- PROGRAMS is where was installed PostgreSQL
- NN is the PostgreSQL version

Create the subfolders data/datasets if don't exist

Creating tables in PostgreSQL – Sales Table data example

- SalesforSQL.csv
- Change the location as per your installation directory
- COPY sales from '/Library/PostgreSQL/14/data/dataset/SalesforSQL.csv' delimiter ',' csv header;
 - Example Linux: COPY sales from '/var/lib/postgresql/13/data/dataset/SalesforSQL.csv' delimiter ',' csv header;
 - Example Windows: COPY sales from 'C:\Program Files\PostgreSQL\14\data\dataset\SalesforSQL.csv' delimiter ',' csv header;
 - Select and run only this line
- To Check if the data has been correctly imported, run the select command
 - SELECT * FROM sales;

Creating tables in PostgreSQL – Sales Table data example

Query Editor Query History Scratches

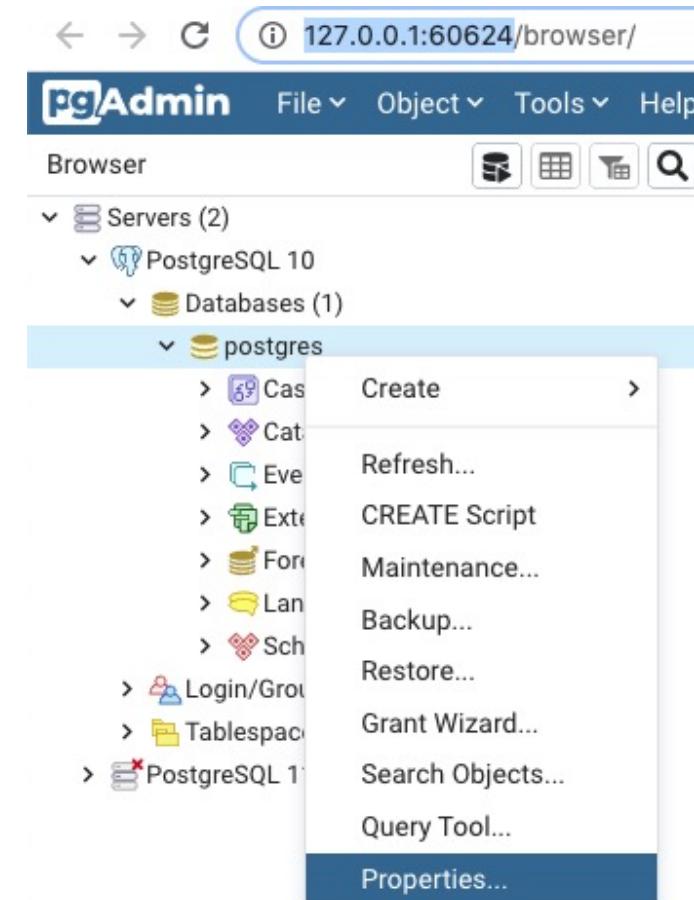
```
1 Create table Sales (
2 Order_Line int primary key,
3 Order_ID varchar,
4 Order_Date date,
5 Ship_Date date,
6 Ship_Mode varchar,
7 Customer_ID varchar,
8 Product_ID varchar,
9 Sales numeric,
10 Quantity int,
11 Discount numeric,
12 Profit numeric
13 );
14
15
16 COPY sales from '/Library/PostgreSQL/10/data/dataset/SalesforSQL.csv' delimiter ',' csv header;
17
18 SELECT * FROM sales;
19
```

Data Output Explain Messages Notifications

	order_line [PK] integer	order_id character varying	order_date date	ship_date date	ship_mode character varying	customer_id character varying	product_id character varying	sales numeric	quantity integer	discount numeric	profit numeric
1	7973	CA-2017-166142	2020-01-01	2020-01-05	Standard Class	MM-17260	OFF-BI-10004094	26.55	3	0	13.01
2	7974	CA-2017-166142	2020-01-01	2020-01-05	Standard Class	MM-17260	FUR-TA-10004607	310.44	3	0.3	-48.78
3	7975	CA-2017-164378	2020-01-01	2020-01-04	Second Class	MM-18055	OFF-AR-10001177	6.56	2	0	1.9
4	7976	CA-2017-164378	2020-01-01	2020-01-04	Second Class	MM-18055	OFF-LA-10000634	7.83	3	0	3.6
5	7977	CA-2017-164378	2020-01-01	2020-01-04	Second Class	MM-18055	TEC-AC-10004708	41.9	2	0	8.8
6	7978	CA-2017-164378	2020-01-01	2020-01-04	Second Class	MM-18055	FUR-CH-10002084	664.15	6	0.1	88.55
7	7979	CA-2017-164378	2020-01-01	2020-01-04	Second Class	MM-18055	OFF-PA-10004519	8.96	2	0	4.39
8	7980	US-2017-128951	2020-01-01	2020-01-03	First Class	RS-19420	OFF-AP-10002191	179.94	3	0	50.38
9	7981	US-2017-128951	2020-01-01	2020-01-03	First Class	RS-19420	FUR-TA-10004575	872.94	3	0	157.13
10	7982	US-2017-128951	2020-01-01	2020-01-03	First Class	RS-19420	OFF-PA-10002177	10.06	2	0	5.00

Connect to database: information

- **Host server address/location:** 127.0.0.1 take a look to your url
<http://127.0.0.1:60624/>
- **Name database:** postgres
- **Password**
- **User:** postgres



Import data from SQL database in PDI

Table input

Step name **Table input**

Connection **SalesDBConnection**

Wizard... (button circled in red)

Get SQL select statement.

SQL

```
SELECT <values> FROM <table name> WHERE <conditions>
```

Line 1 Column 0

Store column info in step meta data

Enable lazy conversion

Replace variables in script?

Insert data from step

Execute for each row?

Limit size

Help OK Preview Cancel

Select the database name and type

Select 'next' to proceed

Name of the database connection	SalesDBConnection
Type of database to connect to	Borland Interbase
	Calpont InfiniDB
	Cloudera Impala
	dBase III, IV or 5
	Exasol 4
	ExtenDB
	Firebird SQL
	Generic database
	Google BigQuery
	Greenplum
	Gupta SQL Base
	H2
	Hadoop Hive
	Hadoop Hive 2/3
	Hive Warehouse Connector
	Hypersonic
	IBM DB2
	Impala
	Infobright
	Informix
	Ingres
	Ingres VectorWise
	Intersystems Cache
	KingbaseES
	LucidDB
	MariaDB
	MaxDB (SAP DB)
	MonetDB
	MS Access
	MS SQL Server
	MS SQL Server (Native)
	MySQL
	Native Mondrian
	Neoview
	Netezza
	Oracle
	Oracle RDB
	Palo MOLAP Server
	Pentaho Data Services
	PostgreSQL
	Redshift
	Remedy Action Request System
	SAP ERP System
	Snowflake
	SparkSQL
	SQLite
	Sybase
	SybaseIQ
	Teradata
	UniVerse database

Type of database access to use

Native (JDBC) (selected)

ODBC

JNDI

Import data from SQL database in PDI

Set the JDBC Settings

Select 'next' to proceed

Host name of the database server

The TCP/IP port

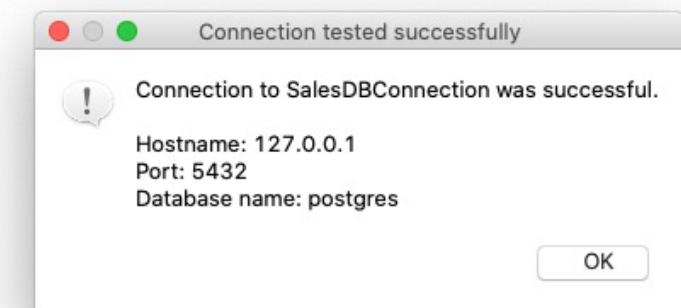
The name of the database

username and password

Click 'Finish' to create the database connection

The username

The password



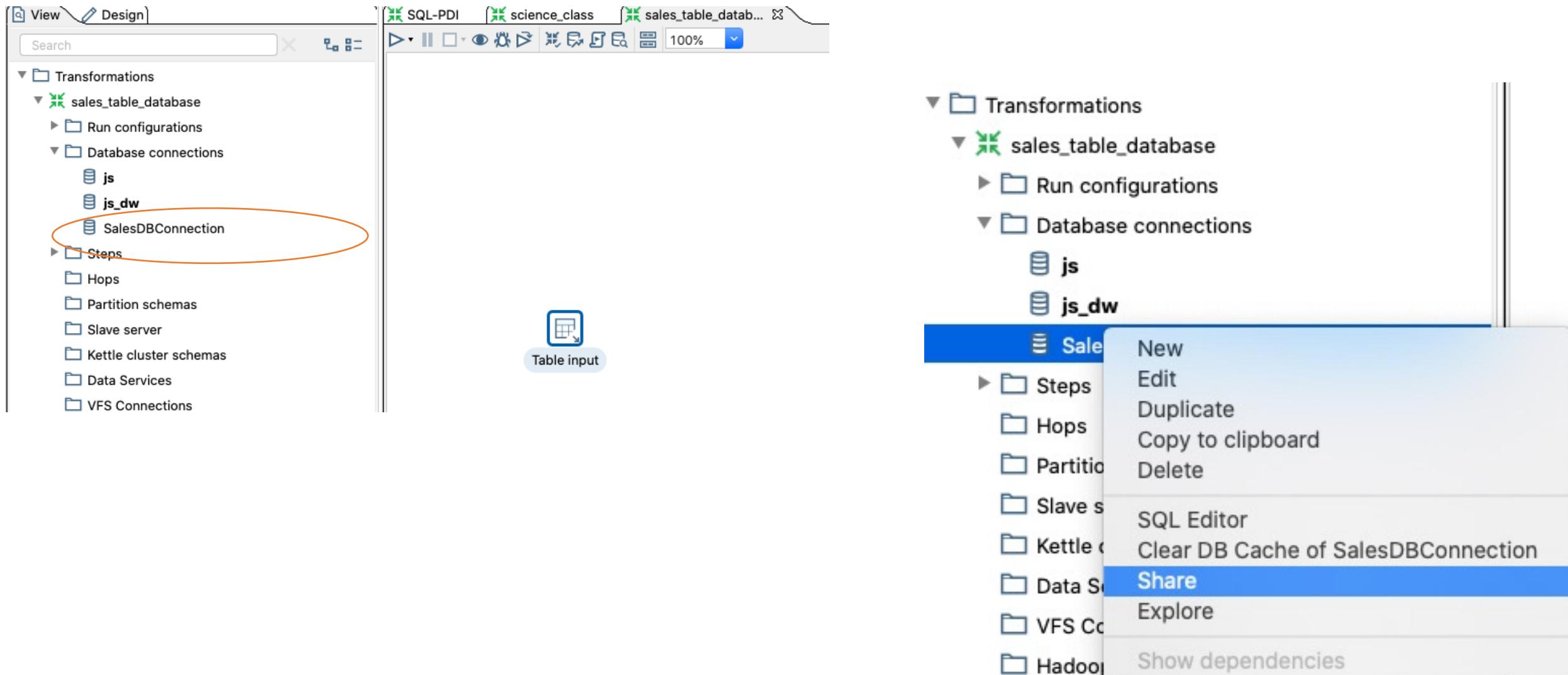
Query data from SQL database in PDI

- SELECT <values> FROM <table name> WHERE <conditions>
- SELECT * FROM sales

sales_table_database.ktr

Rows of step: Table input (1000 rows)												
#	order_line	order_id	order_date	ship_date	ship_mode	customer_id	product_id	sales	quantity	discount	profit	
1	7973	CA-2017-166...	2020/01/01 00:00:0...	2020/01/05 00:00:0...	Standard Class	MM-17260	OFF-BI-10004094	26.55	3	0.0	13.01	
2	7974	CA-2017-166...	2020/01/01 00:00:0...	2020/01/05 00:00:0...	Standard Class	MM-17260	FUR-TA-10004607	310.44	3	0.3	-48.78	
3	7975	CA-2017-164...	2020/01/01 00:00:0...	2020/01/04 00:00:0...	Second Class	MM-18055	OFF-AR-10001...	6.56	2	0.0	1.9	
4	7976	CA-2017-164...	2020/01/01 00:00:0...	2020/01/04 00:00:0...	Second Class	MM-18055	OFF-LA-10000634	7.83	3	0.0	3.6	
5	7977	CA-2017-164...	2020/01/01 00:00:0...	2020/01/04 00:00:0...	Second Class	MM-18055	TEC-AC-10004...	41.9	2	0.0	8.8	
6	7978	CA-2017-164...	2020/01/01 00:00:0...	2020/01/04 00:00:0...	Second Class	MM-18055	FUR-CH-10002...	664.15	6	0.1	88.55	
7	7979	CA-2017-164...	2020/01/01 00:00:0...	2020/01/04 00:00:0...	Second Class	MM-18055	OFF-PA-100045...	8.96	2	0.0	4.39	
8	7980	US-2017-128...	2020/01/01 00:00:0...	2020/01/03 00:00:0...	First Class	RS-19420	OFF-AP-100021...	179.94	3	0.0	50.38	
9	7981	US-2017-128...	2020/01/01 00:00:0...	2020/01/03 00:00:0...	First Class	RS-19420	FUR-TA-10004575	872.94	3	0.0	157.13	
10	7982	US-2017-128...	2020/01/01 00:00:0...	2020/01/03 00:00:0...	First Class	RS-19420	OFF-PA-100031...	12.96	2	0.0	6.22	
11	7983	US-2017-156...	2020/01/02 00:00:0...	2020/01/04 00:00:0...	Second Class	SF-20065	FUR-CH-10002...	71.37	2	0.3	-1.02	
12	7984	CA-2017-109...	2020/01/02 00:00:0...	2020/01/07 00:00:0...	Standard Class	VM-21685	OFF-AR-10003...	2.91	2	0.2	0.91	
13	7985	US-2017-152...	2020/01/02 00:00:0...	2020/01/09 00:00:0...	Standard Class	NF-18385	FUR-CH-10004...	242.35	3	0.2	15.15	
14	7986	CA-2017-139...	2020/01/03 00:00:0...	2020/01/08 00:00:0...	Standard Class	SW-20455	FUR-FU-10002...	7.9	2	0.2	2.17	
15	7987	US-2017-105...	2020/01/03 00:00:0...	2020/01/09 00:00:0...	Standard Class	BE-11335	TEC-PH-10004...	269.98	2	0.0	67.5	
16	7988	US-2017-105...	2020/01/03 00:00:0...	2020/01/09 00:00:0...	Standard Class	BE-11335	OFF-PA-100043...	99.9	5	0.0	47.95	
17	7989	US-2017-105...	2020/01/03 00:00:0...	2020/01/09 00:00:0...	Standard Class	BE-11335	FUR-FU-10004...	39.08	4	0.0	14.46	
18	7990	CA-2017-126...	2020/01/03 00:00:0...	2020/01/07 00:00:0...	Standard Class	AB-10255	TEC-CO-10004...	479.98	2	0.2	90.0	
19	7991	CA-2017-142...	2020/01/03 00:00:0...	2020/01/05 00:00:0...	Second Class	AJ-10795	OFF-PA-100046...	32.4	5	0.0	15.55	
20	7992	CA-2017-142...	2020/01/03 00:00:0...	2020/01/05 00:00:0...	Second Class	AJ-10795	OFF-EN-10002...	57.9	5	0.0	28.95	
21	7993	CA-2017-142...	2020/01/03 00:00:0...	2020/01/05 00:00:0...	Second Class	AJ-10795	OFF-ST-10002957	10.56	2	0.0	0.0	
22	7994	CA-2017-142...	2020/01/03 00:00:0...	2020/01/05 00:00:0...	Second Class	AJ-10795	FUR-BO-10002...	1194.17	5	0.15	210.74	
23	7995	US-2017-142...	2020/01/04 00:00:0...	2020/01/09 00:00:0...	Standard Class	ML-17410	FUR-TA-10001932	801.6	5	0.5	-448.9	
24	7996	US-2017-142...	2020/01/04 00:00:0...	2020/01/09 00:00:0...	Standard Class	ML-17410	FUR-CH-10004...	161.57	2	0.2	10.1	
25	7997	US-2017-142...	2020/01/04 00:00:0...	2020/01/09 00:00:0...	Standard Class	ML-17410	OFF-PA-100002...	16.1	2	0.2	5.23	
26	7998	US-2017-142...	2020/01/04 00:00:0...	2020/01/09 00:00:0...	Standard Class	ML-17410	OFF-BI-10003350	7.66	4	0.7	-6.12	
27	7999	US-2017-142...	2020/01/04 00:00:0...	2020/01/09 00:00:0...	Standard Class	MI-17410	FUR-CH-100000	311.98	2	0.2	-42.9	

Global database connection



SQL-PDI connection: a simple example

1. Create a new table in postgresql by running this command in PGAdmin

```
create table science_class(  
Enrollment_no INT,  
Name VARCHAR,  
Science_Marks INT  
);
```

2. Insert some sample values

```
insert into science_class values (1,'Popeye',33);  
insert into science_class values (2,'Olive',54);  
insert into science_class values (3,'Brutus',98);
```

3. Query: [`select * from science_class`](#)

SQL-PDI connection: task examples

1. Read:

- Retrieve all data from the table ‘Science_Class’

2. Read with conditions

- Retrieve the name of students who have scored more than 60 marks

3. Update

- Update the marks of Popeye to 45

4. Insert

- Insert a new row with “Wimpy” who has scored 75 marks

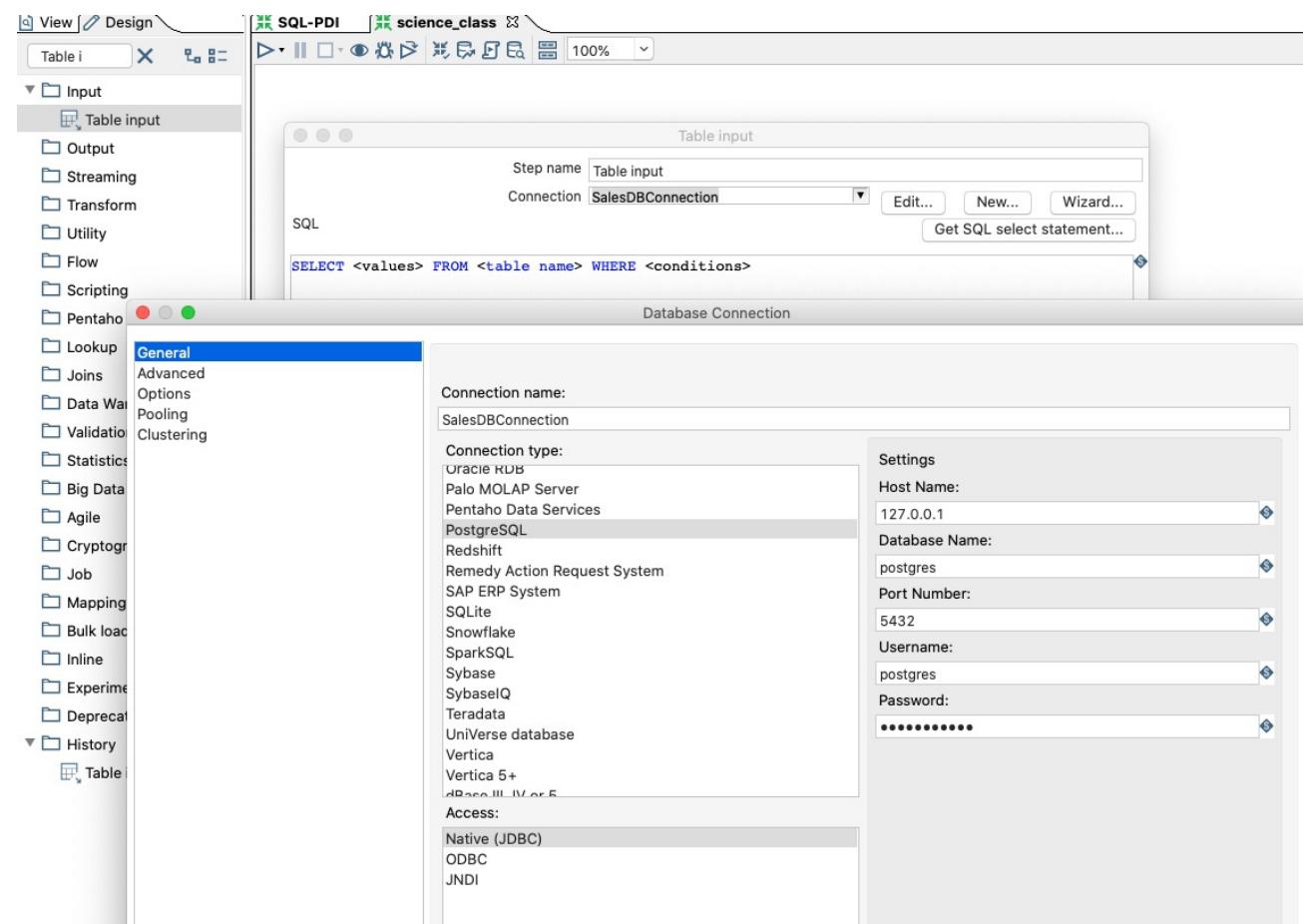
5. Delete

- Delete the record of Wimpy

SQL-PDI connection: Read example

Retrieve all data from the table
'Science_Class'

- Input table -> reuse database connection
- **SELECT * FROM science_class**



SQL-PDI connection: Read with conditions example

- Retrieve the name of students who have scored more than 60 marks
- `SELECT * FROM science_class WHERE science_mark > 60`

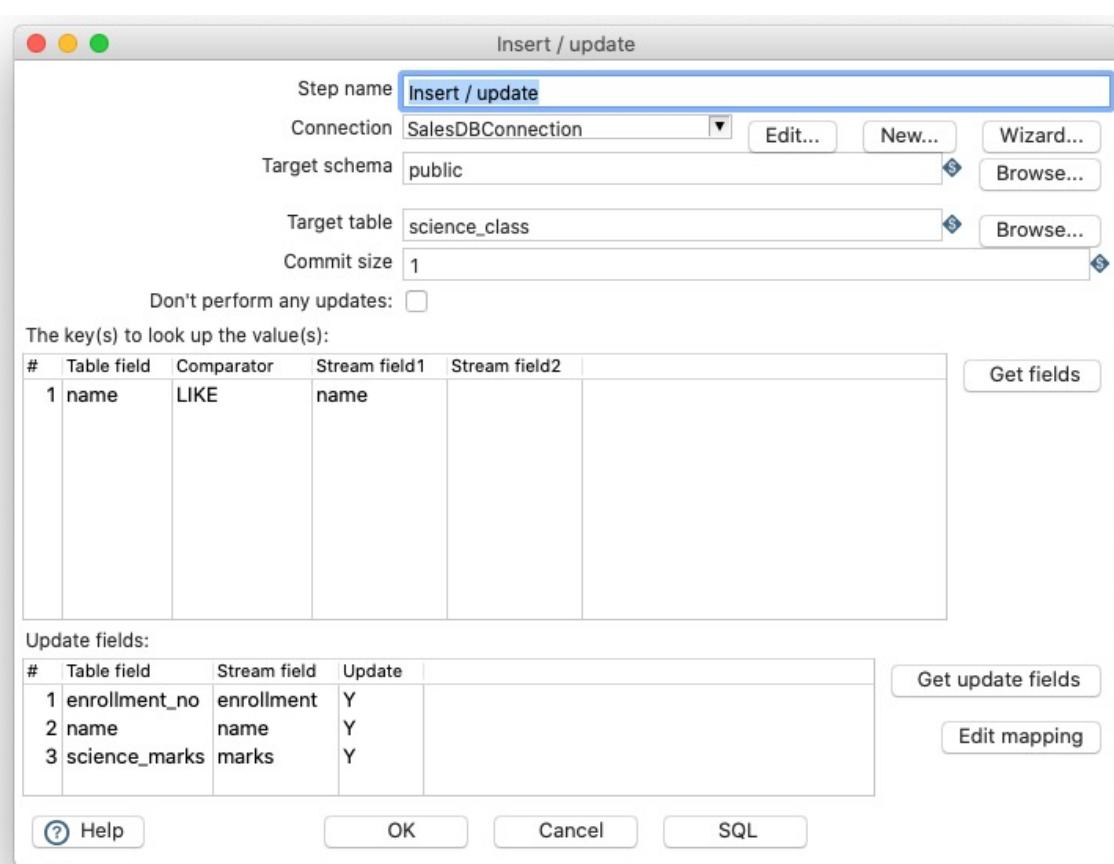
enrollment_no	name	science_marks
3	Brutus	98

science_class_read.ktr

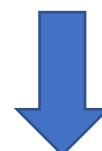
SQL-PDI connection: Update and Insert example

- Update the marks of Popeye to 45
 - Insert a new row with “Wimpy” who has scored 75 marks

science class update insert.ktr



Data Output		Explain	Messages	Notifications	
	enrollment_no integer	name character varying		science_marks integer	
1		1	Popeye		33
2		2	Olive		54
3		3	Brutus		98



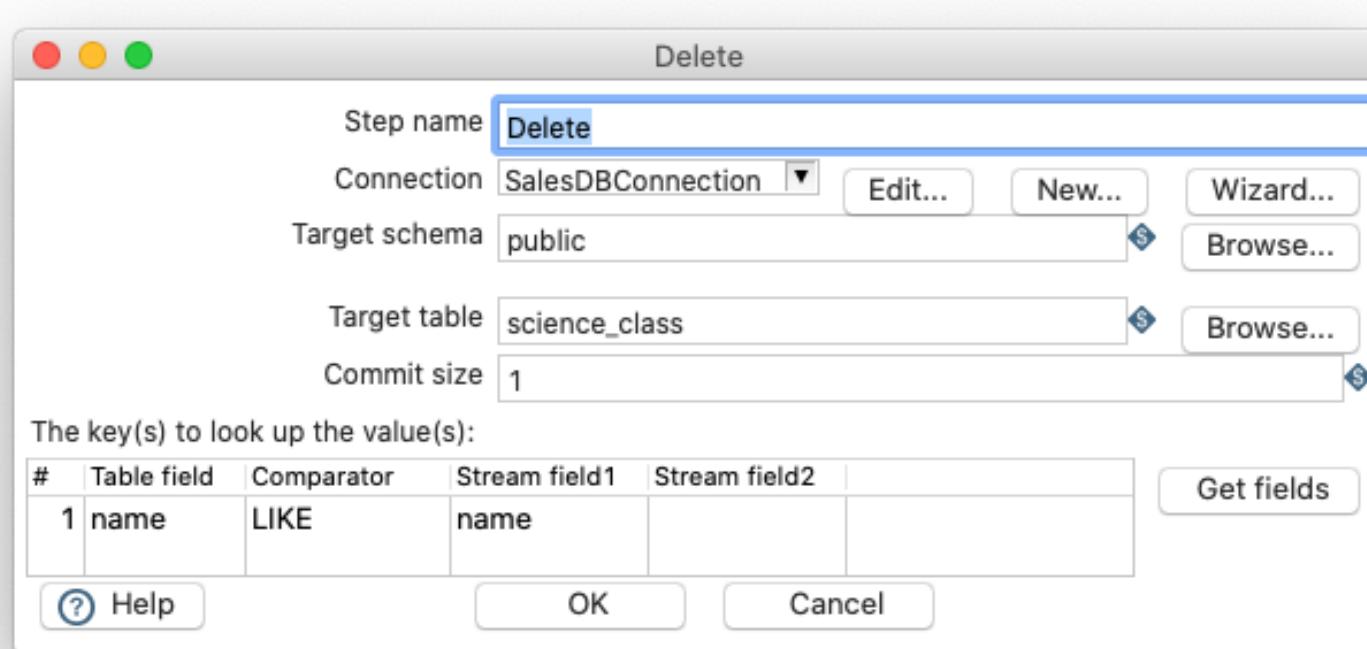
	enrollment_no	name	science_marks
1	integer	Olive	54
2	3	Brutus	98
3	1	Popeye	45
4	4	Wimpy	75

SQL-PDI connection: Delete example

- Delete the record of Wimpy



	enrollment_no integer	name character varying	science_marks integer
1		Olive	54
2		Brutus	98
3		Popeye	45



science_class_delete.ktr

Loading Data

Loading Data

Run SQL create table query on PgAdmin

- Product table
- Customer table
- Final sales table

Creating Product table

```
create table product (
    surr_id int primary key, ←
    product_id varchar default 'N/A' NOT NULL,
    category varchar default 'N/A' NOT NULL,
    sub_category varchar default 'N/A' NOT NULL,
    product_name varchar default 'N/A' NOT NULL,
    start_date date,
    end_date date,
    version int default 1 NOT NULL,
    current varchar default 'Y' NOT NULL,
    lastupdate date
);
```

Surrogate key used to reference the information in the product

Type 2 SCD (Slowly changing dimensions)

select * from product

Creating Customer table

```
create table customer (
    surr_id int primary key, ←
    customer_id varchar default 'N/A' NOT NULL,
    customer_name varchar default 'N/A' NOT NULL,
    segment varchar default 'N/A' NOT NULL,
    age int default '0' NOT NULL,
    city varchar default 'N/A' NOT NULL,
    state_name varchar default 'N/A' NOT NULL,
    country varchar default 'N/A' NOT NULL,
    postal_code varchar default 'N/A' NOT NULL,
    region varchar default 'N/A' NOT NULL
);
```

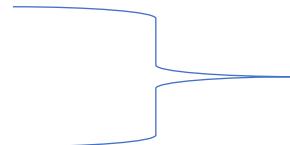
Surrogate key used to reference the information in the customer

Type 1 SCD (Slowly changing dimensions)

select * from customer

Creating Final sales table

```
create table finalsales(  
    order_line int primary key,  
    order_id varchar default 'N/A' NOT NULL,  
    order_date date default '1900-01-01' NOT NULL,  
    ship_date date default '1900-01-01' NOT NULL,  
    ship_mode varchar default 'N/A' NOT NULL,  
    s_cust_id int default '0' NOT NULL,  
    s_prod_id int default '0' NOT NULL,  
    sales numeric default '0' NOT NULL,  
    quantity int default '0' NOT NULL,  
    discount numeric default '0' NOT NULL,  
    profit numeric default '0' NOT NULL  
);
```



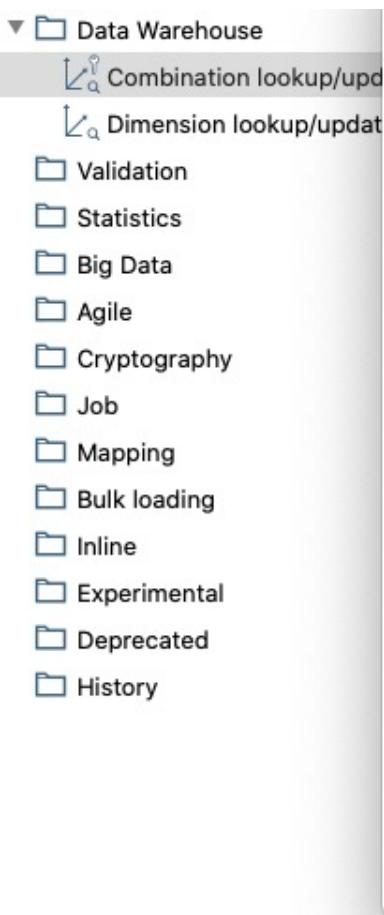
Instead of having the business reference keys are used Surrogate keys

```
select * from finalsales
```

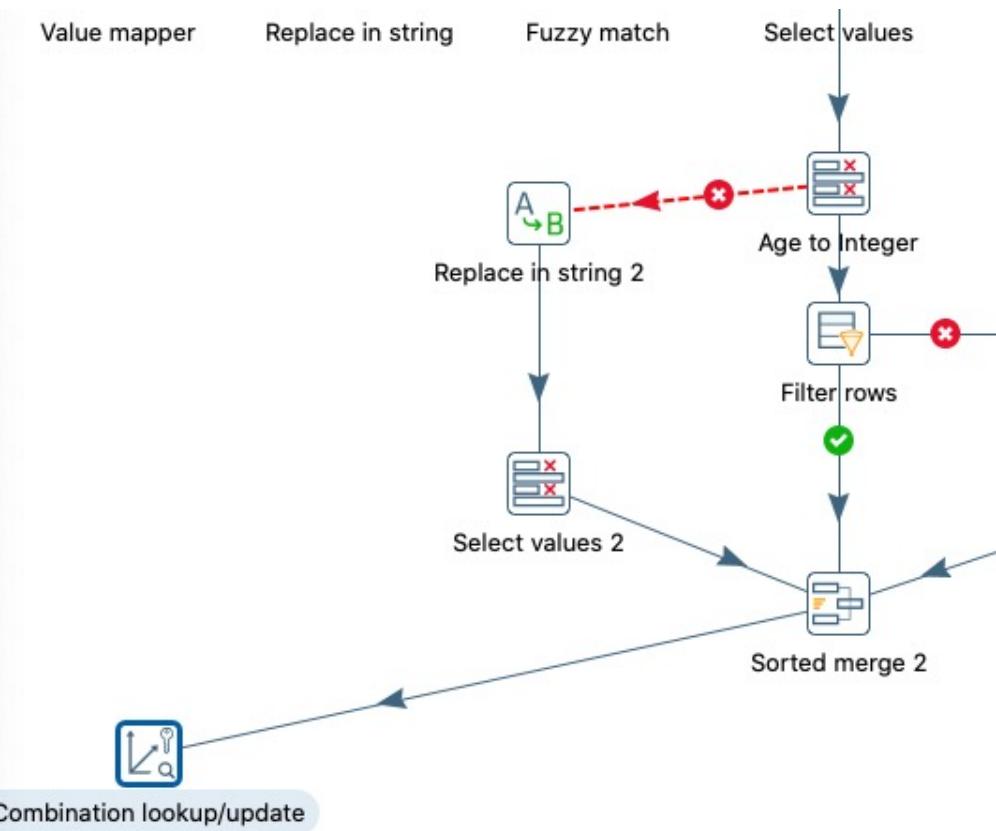
Loading Data: Customer table

- We want to check **whether a particular customer information already exists or not**
 - if it **already exists** we want to update it
 - if it **does not**, we want to insert a new row for that particular customer

Loading Data: Customer table



select * from customer



CustomerTransformation_check5_ded
clean_fuzzy_Validation_errorh_Load.ktr

Loading Data: Customer table

select * from customer

	surr_id [PK] integer	customer_id character varying	customer_name character varying	segment character varying	age integer	city character varying	state_name character varying	country character varying	postal_code character varying	region character varying
1		1 AA-10315	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A
2		2 AA-10375	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A
3		3 AA-10480	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A
4		4 AA-10645	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A
5		5 AB-10015	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A
6		6 AB-10060	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A
7		7 AB-10105	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A
8		8 AB-10150	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A
9		9 AB-10165	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A
10		10 AB-10255	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A
11		11 AB-10600	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A
12		12 AC-10420	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A
13		13 AC-10450	N/A	N/A	0	N/A	N/A	N/A	N/A	N/A

Loading Data: Customer table

Output

- Insert / update
- LDAP output
- Salesforce update
- Salesforce upsert
- Synchronize after merge
- Update**
- Streaming
- Transform
- Utility
- Flow
- Scripting
- Pentaho Server
- Lookup
- Joins

Data Warehouse

- Combination lookup/update
- Dimension lookup/update

Validation

Statistics

Big Data

Agile

Cryptography

Job

Mapping

Bulk loading

Inline

Experimental

Sort rows 3 **Sorted merge** Unique rows

Step name: **Update**

Connection: SalesDBConnection Edit... New... Wizard...

Target schema: public

Target table: customer

Commit size: 1

Use batch updates?

Skip lookup

Ignore lookup failure? Flag field (key found)

The key(s) to look up the value(s):

#	Table field	Comparator	Stream field1	Stream field2
1	customer_id	=	Customer ID	

Get fields

Update fields:

#	Table field	Stream field
1	customer_name	Customer Name
2	segment	Segment
3	age	Age
4	city	City
5	state_name	State
6	country	Country
7	postal_code	Postal Code
8	region	Region

Get update fields

value mapper Replace in string Fuzzy match Select values

Metrics

Preview data

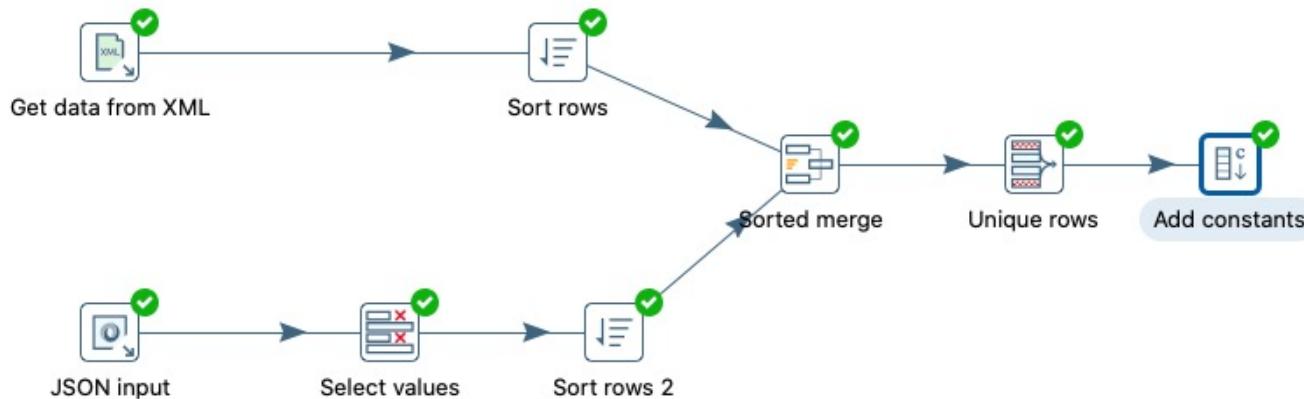
220, O=0, R=0, W=220, U=0, E=0
 =220, U=0, E=0
 W=793, U=0, E=0
 V=793, U=0, E=0
 W=793, U=0, E=0
 93, W=793, U=0, E=0
 N=793, U=0, E=0
 W=793, U=0, E=0

Loading Data: Customer table

```
select * from customer;
```

	surr_id [PK] integer	customer_id character varying	customer_name character varying	segment character varying	age integer	city character varying	state_name character varying	country character varying	postal_code character varying	region character varying
1	2	AA-10375	Allen Armold	Consumer	22	Mesa	Arizona	United States	85204	West
2	4	AA-10645	Anna Andreadi	Consumer	32	Chester	Pennsylvania	United States	19013	East
3	6	AB-10060	Adam Bellavance	Home Office	25	New York City	New York	United States	10009	East
4	7	AB-10105	Adrian Barton	Consumer	63	Phoenix	Arizona	United States	85023	West
5	9	AB-10165	Alan Barnes	Consumer	22	Los Angeles	California	United States	90036	West
6	11	AB-10600	Ann Blume	Corporate	34	Tucson	Arizona	United States	85705	West
7	12	AC-10420	Alyssa Crouse	Corporate	69	San Francisco	California	United States	94122	West
8	26	AG-10765	Anthony Garverick	Home Office	40	Philadelphia	Pennsylvania	United States	19120	East
9	27	AG-10900	Arthur Gainer	Consumer	56	Tucson	Arizona	United States	85705	West
10	28	AH-10030	Aaron Hawkins	Corporate	60	Philadelphia	Pennsylvania	United States	19134	East
11	29	AH-10075	Adam Hart	Corporate	21	New York City	New York	United States	10011	East
12	30	AH-10120	Adrian Hane	Home Office	27	Tucson	Arizona	United States	85705	West
13	31	AH-10195	Alan Haines	Corporate	67	Tamarac	Florida	United States	33319	South
14	32	AH-10210	Alan Hwang	Consumer	58	Brentwood	California	United States	94513	West
15	33	AH-10465	Amy Hunt	Consumer	24	New York City	New York	United States	10035	East
16	34	AH-10585	Angele Hood	Consumer	34	Chicago	Illinois	United States	60623	Central

Loading Data: Product table



Type 2 SCD (Slowly changing dimensions)
Maintain historical data

Execution Results

Logging (i) Execution History (1) Step Metrics (1) Performance Graph (E) Metrics (E) Preview data

First rows Last rows Off

#	Product_ID	Category	Sub_Category	Product_Name	effective date
1	FUR-BO-10000...	Furniture	Bookcases	Bush Birmingham Collection Bookcase Dark Cherry	10/10/2020
2	FUR-BO-10000...	Furniture	Bookcases	Sauder Camden County Barrister Bookcase Planked Cherry Finish	10/10/2020
3	FUR-BO-10000...	Furniture	Bookcases	Sauder Indalewood Library Bookcases	10/10/2020

Add constants

Step name Add constants

Fields :

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Value	Set empty string?
1	effective date	Date	dd/MM/yyyy						10/10/20...	N

OK Cancel

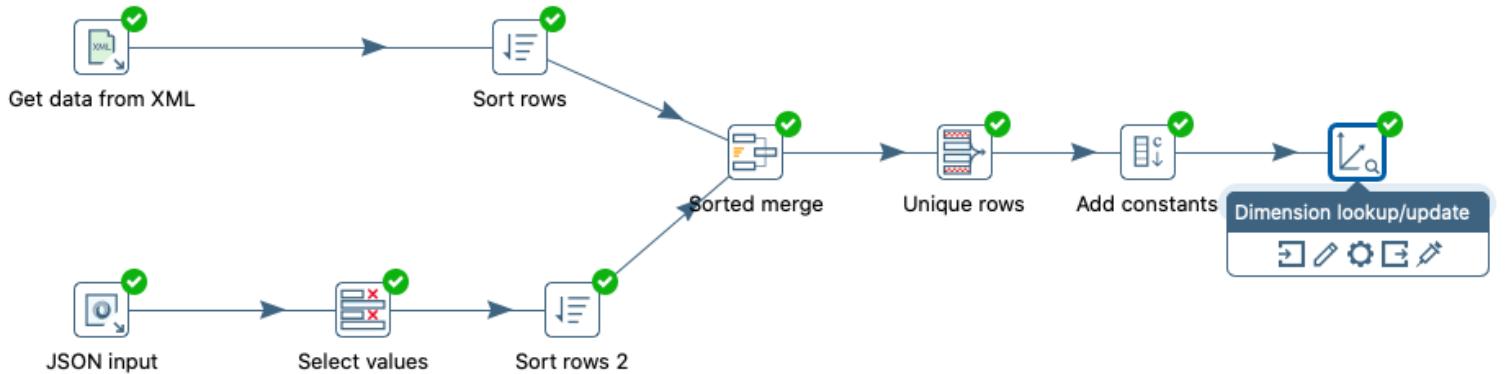
Help

14 FUR-BO-10001... Furniture BOOKCASES Atlantic Metals Mobile 5-Shelf Bookcases Custom Colors

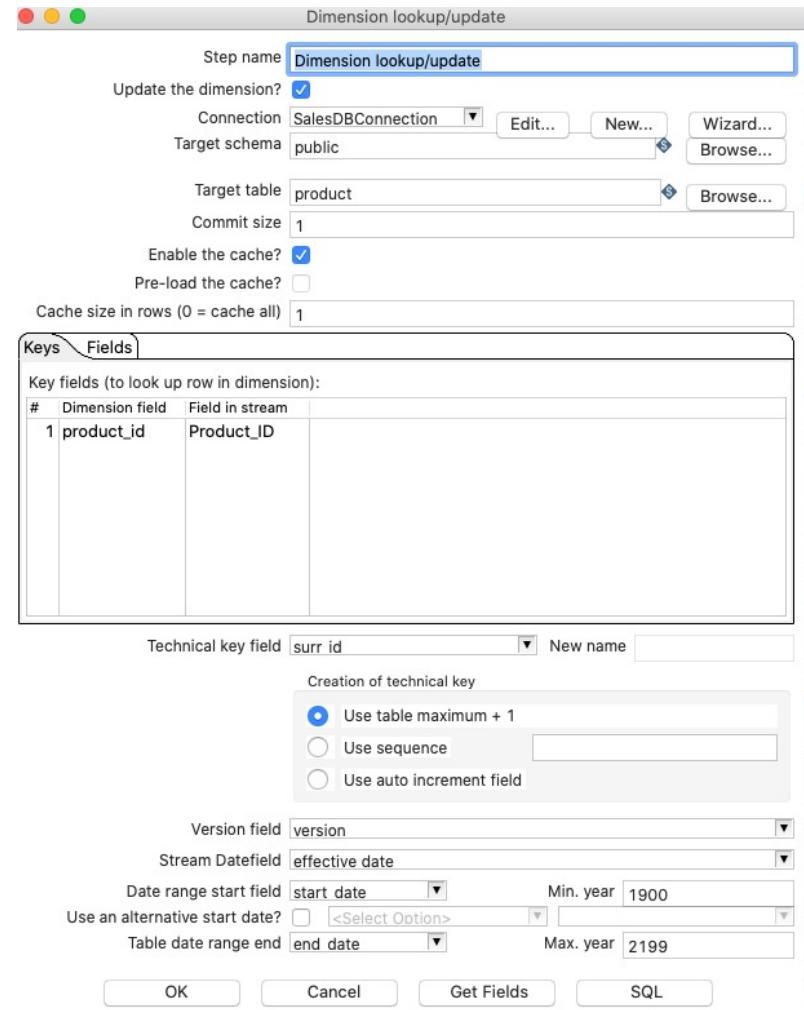
Type 2 SCD (Slowly changing dimensions)
Maintain historical data

Loading Data: Product table

ProductTransformation_1_clean_load.ktr



Keys			
Fields			
Lookup/Update fields			
#	Dimension field	Stream field to compare with	Type of dimension update
1	category	Category	Insert
2	sub_category	Sub_Category	Insert
3	product_name	Product_Name	Insert
4	current		Last version (without stream field as source)
5	lastupdate	effective date	Update



Loading Data: Product table

```
select * from product;
```

Data Output												Explain		Messages		Notifications	
#	surr_id [PK] integer	product_id character varying	category character varying	sub_category character varying	product_name character varying	start_date date	end_date date	version integer	current character varying	lastupdate date							
1	0	N/A	N/A	N/A	N/A	[null]	[null]	1	Y	[null]							
2	1	FUR-BO-10000112	Furniture	Bookcases	Bush Birmingham Coll...	1900-01-01	2199-12-31	1	Y	2020-10-10							
3	2	FUR-BO-10000330	Furniture	Bookcases	Sauder Camden Count...	1900-01-01	2199-12-31	1	Y	2020-10-10							
4	3	FUR-BO-10000362	Furniture	Bookcases	Sauder Inglewood Libr...	1900-01-01	2199-12-31	1	Y	2020-10-10							
5	4	FUR-BO-10000468	Furniture	Bookcases	OSullivan 2-Shelf Heav...	1900-01-01	2199-12-31	1	Y	2020-10-10							
6	5	FUR-BO-10000711	Furniture	Bookcases	Hon Metal Bookcases ...	1900-01-01	2199-12-31	1	Y	2020-10-10							
7	6	FUR-BO-10000780	Furniture	Bookcases	OSullivan Plantations 2...	1900-01-01	2199-12-31	1	Y	2020-10-10							
8	7	FUR-BO-10001337	Furniture	Bookcases	OSullivan Living Dimen...	1900-01-01	2199-12-31	1	Y	2020-10-10							
9	8	FUR-BO-10001519	Furniture	Bookcases	OSullivan 3-Shelf Heav...	1900-01-01	2199-12-31	1	Y	2020-10-10							
10	9	FUR-BO-10001567	Furniture	Bookcases	Bush Westfield Collecti...	1900-01-01	2199-12-31	1	Y	2020-10-10							
11	10	FUR-BO-10001601	Furniture	Bookcases	Sauder Mission Library...	1900-01-01	2199-12-31	1	Y	2020-10-10							
12	11	FUR-BO-10001608	Furniture	Bookcases	Hon Metal Bookcases ...	1900-01-01	2199-12-31	1	Y	2020-10-10							

Loading Data: Sales table

Instead of having the business reference keys
are used Surrogate keys

Database lookup

Step name: Database lookup 2

Connection: SalesDBConnection

Lookup schema: public

Lookup table: customer

Enable cache?

Cache size in rows (0=cache everything): 0

Load all data from table

The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	customer_id	=	Customer_ID	

Values to return from the lookup table:

#	Field	New name	Default	Type
1	surr_id	s_cust_id	None	

Do not pass the row if the lookup fails

Fail on multiple results?

Order by

Help OK Cancel Get Fields Get lookup fields

view data

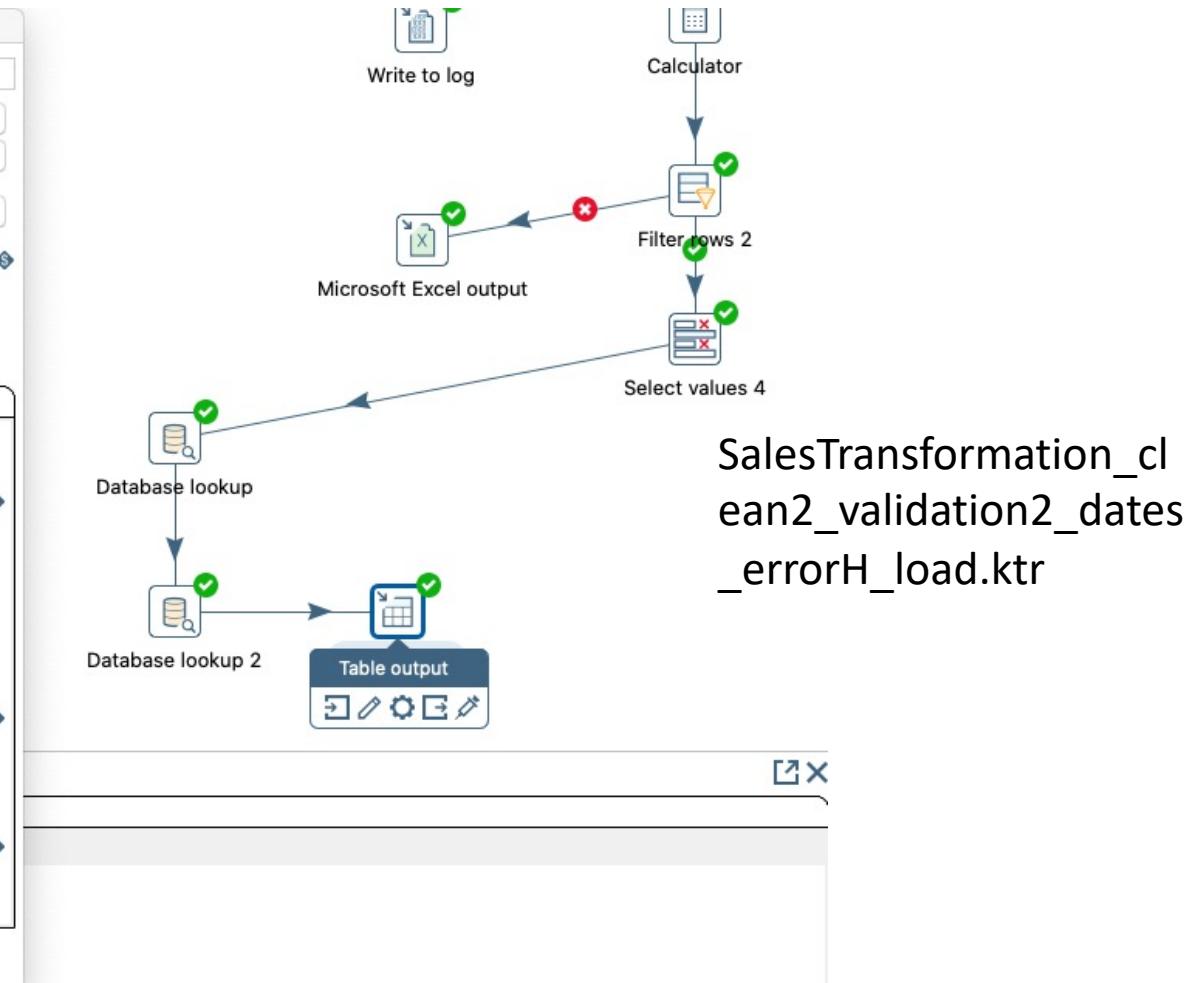
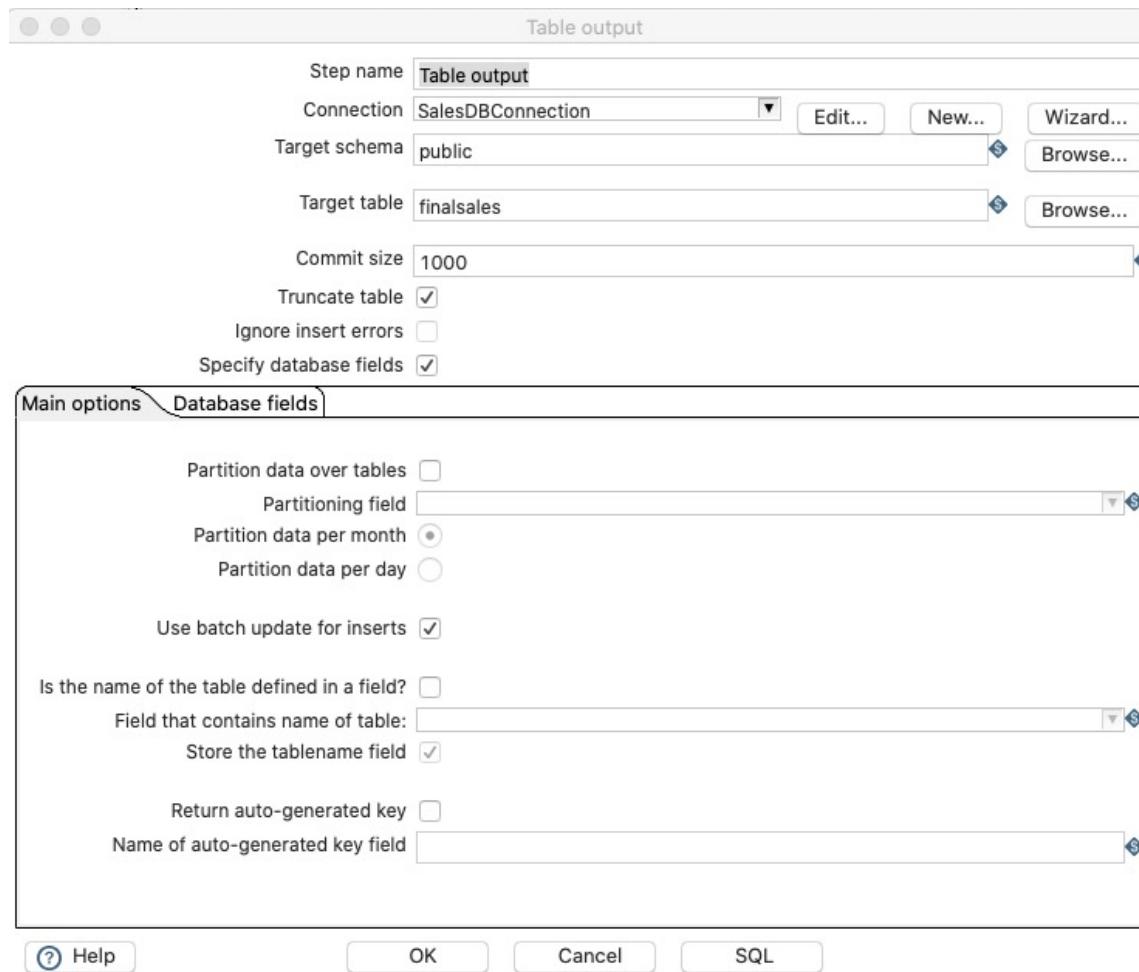
```
graph TD; A[Database lookup] --> B[Write to log]; A --> C[Calculator]; A --> D[Filter rows 2]; A --> E[Microsoft Excel output]; A --> F[Select values 4]; A --> G[Database lookup]
```

Order_ID	Sales	Quantity	Discount	Profit	s_prod_id	s_cust_id
PA-100001...	16.4	2	20	5.5	1027	231
LA-10003223	11.8	3	20	4.3	988	606
ST-10002743	272.7	3	20	-64.8	1379	606
BI-10004094	3.5	2	80	-5.5	805	606
AR-10003...	19.5	3	20	4.9	583	486
OFF-PA-100020...	19.4	3	0	9.3	1150	475
OFF-AR-10002...	12.8	3	0	5.2	553	397
FUR-CH-10004...	2573.8	9	0	746.4	119	498
OFF-BI-10004632	610	2	0	274.5	832	498
OFF-AR-10001...	5.5	2	0	1.5	526	498

SalesTransformation_clean2_validation2_dates_errorH_load.ktr

Loading Data: Sales table

Instead of having the business reference keys are used Surrogate keys



Loading Data: Sales table

```
select * from finalsales;
```

	order_line [PK] integer	order_id character varying	order_date date	ship_date date	ship_mode character varying	s_cust_id integer	s_prod_id integer	sales numeric	quantity integer	discount numeric	profit numeric
1	1	CA-2014-103800	2016-06-21	2016-06-25	Standard Class	231	1027	16.45	2	20	5.55
2	2	CA-2014-112326	2016-06-22	2016-06-26	Standard Class	606	988	11.78	3	20	4.27
3	3	CA-2014-112326	2016-06-22	2016-06-26	Standard Class	606	1379	272.74	3	20	-64.77
4	4	CA-2014-112326	2016-06-22	2016-06-26	Standard Class	606	805	3.54	2	80	-5.49
5	5	CA-2014-141817	2016-06-23	2016-06-30	Standard Class	486	583	19.54	3	20	4.88
6	6	CA-2014-130813	2016-06-24	2016-06-26	Second Class	475	1150	19.44	3	0	9.33
7	7	CA-2014-106054	2016-06-24	2016-06-25	First Class	397	553	12.78	3	0	5.24
8	8	CA-2014-167199	2016-06-24	2016-06-28	Standard Class	498	119	2573.82	9	0	746.41
9	9	CA-2014-167199	2016-06-24	2016-06-28	Standard Class	498	832	609.98	2	0	274.49
10	10	CA-2014-167199	2016-06-24	2016-06-28	Standard Class	498	526	5.48	2	0	1.48
11	11	CA-2014-167199	2016-06-24	2016-06-28	Standard Class	498	1862	391.98	2	0	113.67
12	12	CA-2014-167199	2016-06-24	2016-06-28	Standard Class	498	1846	755.96	4	0	204.11
13	13	CA-2014-167199	2016-06-24	2016-06-28	Standard Class	498	921	31.12	4	0	0.31