

# DATA ANALYTICS (Data Warehouse) Pentaho Data Integration

Luca Cinelli, PhD  
[luca.cinelli@unical.it](mailto:luca.cinelli@unical.it)

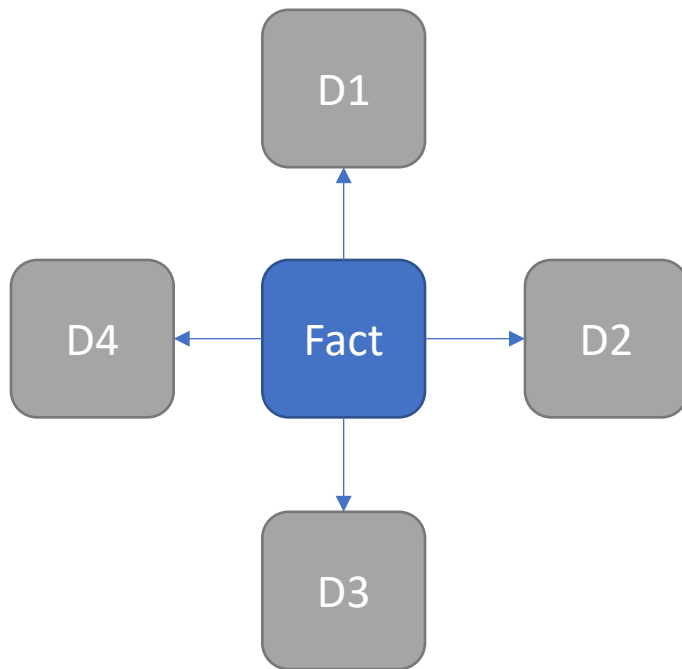
# From data to Dashboards

# From Data to Data Warehouse

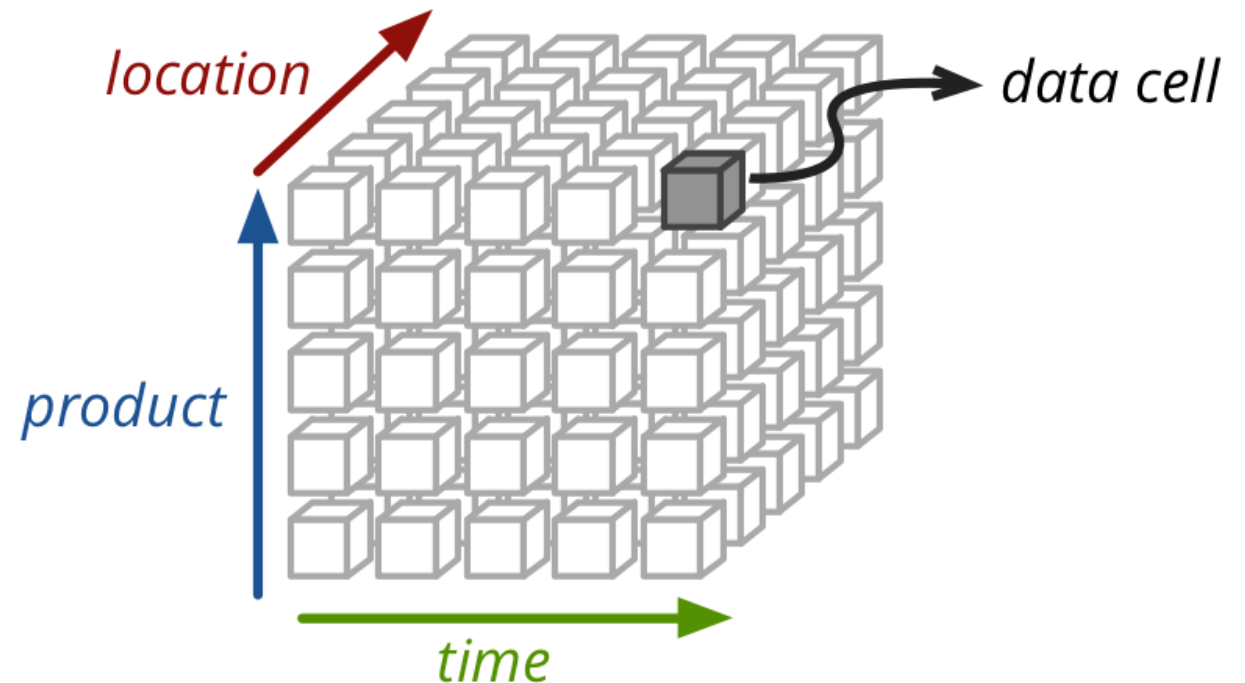


# From Star Schema to Data cube

The process of creation the Star Schema is accomplished in PDI

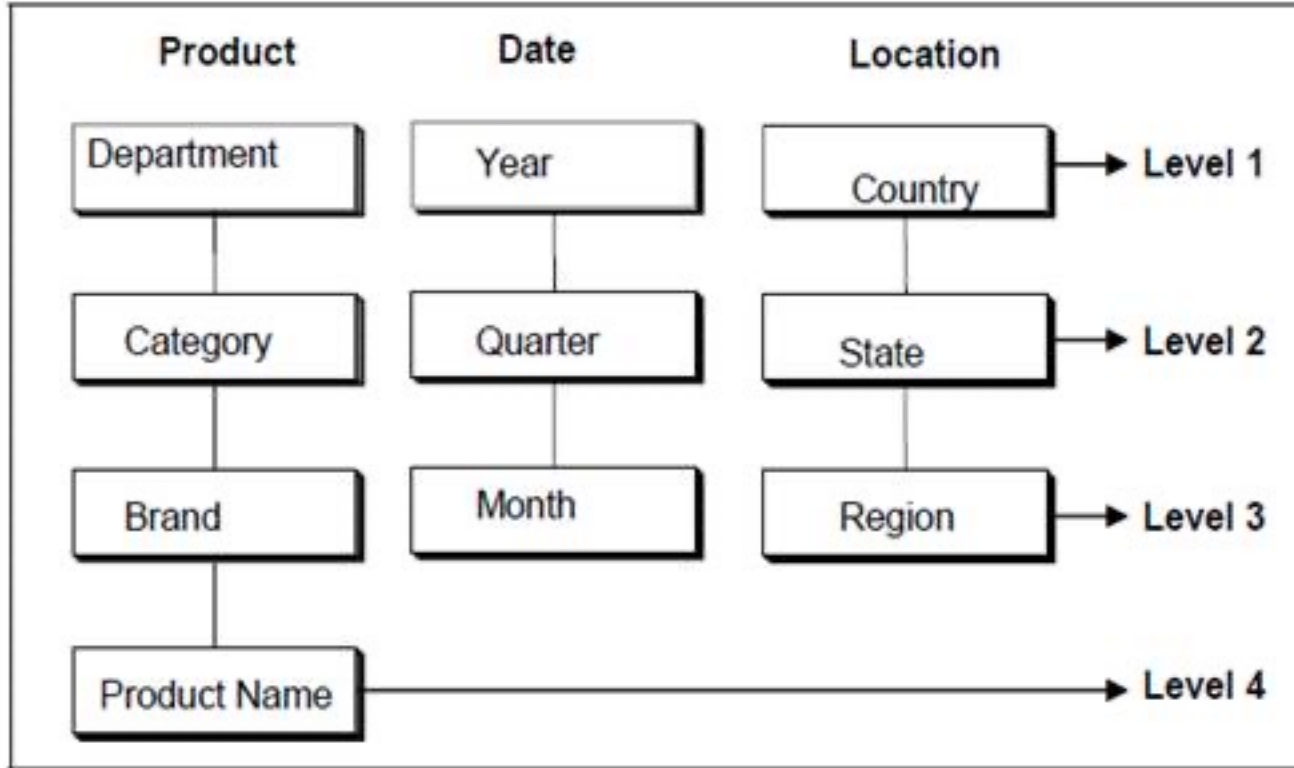


The creation of OnLine Analytical process (OLAP) Cube happens in Mondrian



OLAP Cube is a cube assembled from the Star Schema, where the dimensions are the sides and the measurements are the center.

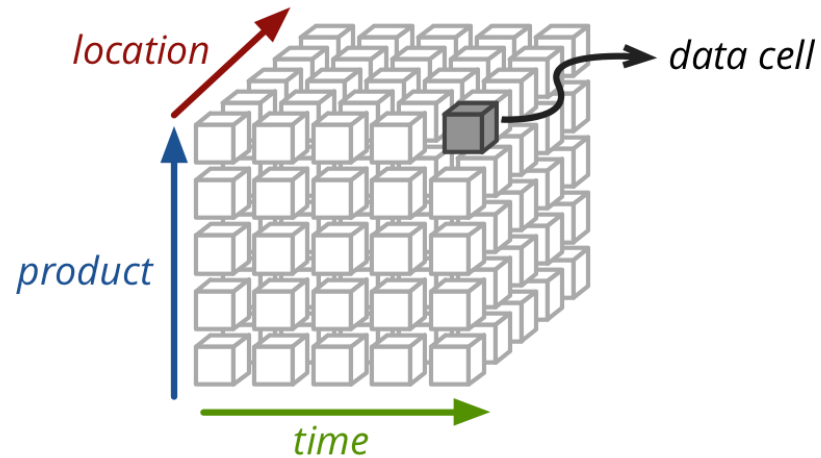
# Hierarchy of Dimensions



More levels into dimensions enables us to make a deeper analysis and dashboard allowing, for example, more filters

# From Data cube to Dashboards

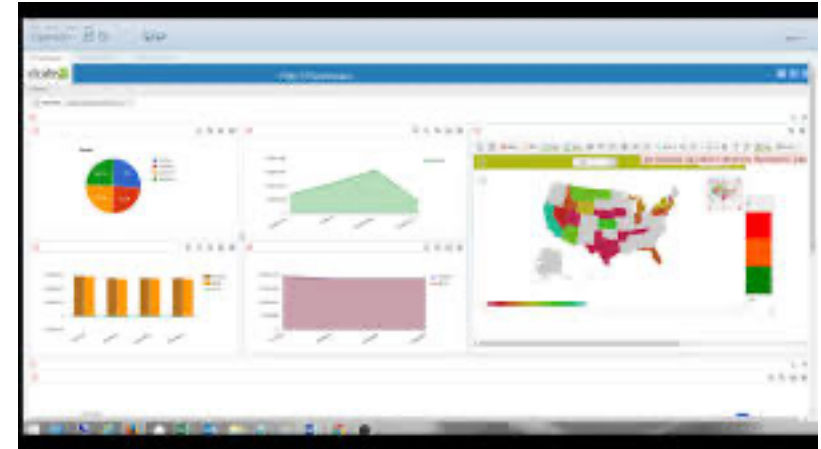
OLAP Cube



publish



Dashboard  
making graphs, tables, and filters

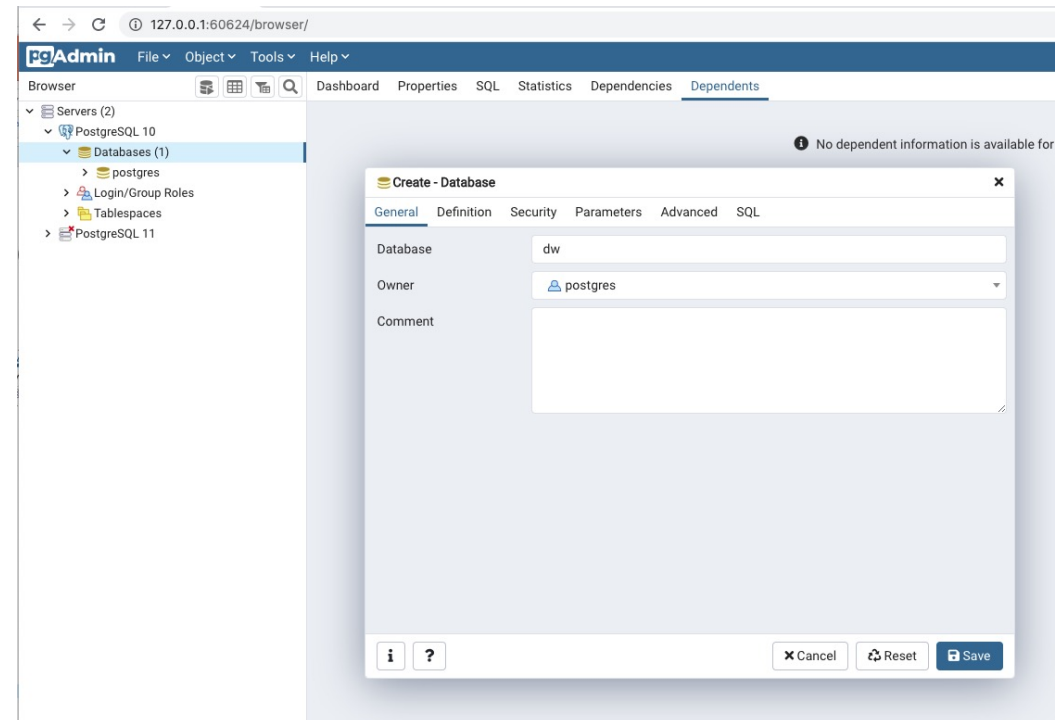


# Installed ETL tool + Database

- We have already installed
  - Pentaho Data Integration to perform our ETL process
  - PostgreSQL to store our Data Warehouse and its pgAdmin graphical interface

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

Create new database: dw



# Download and install Mondrian

https://sourceforge.net/projects/mondrian/files/schema%20workbench/

**SOURCEFORGE**

Open Source Software Business Software Resources

Home / Browse / Business & Enterprise / Enterprise / Mondrian / Files

## Mondrian

Brought to you by: [jhyde](#), [lucboudreau](#), [mbatchelor](#), [pedrofvteixeira](#), [pmgalves](#)

Summary **Files** Reviews Support Wiki News Donate Mailing Lists

**Download Latest Version**  
mondrian-3.14.0.0-12.jar (3.5 MB) **Get Updates**

Home / schema workbench

Name	Modified	Size	Downloads / Week
↶ Parent folder			
3.14.0	2017-05-22	104	104
3.13.0	2016-11-09	10	10
3.12.0	2016-04-13	6	6
3.11.0	2015-12-10	5	5
3.10.0	2015-06-15	0	0
3.9.0	2015-02-16	0	0

<https://sourceforge.net/projects/mondrian/files/schema%20workbench/>

psw-ce-3.14.0.0-12.zip

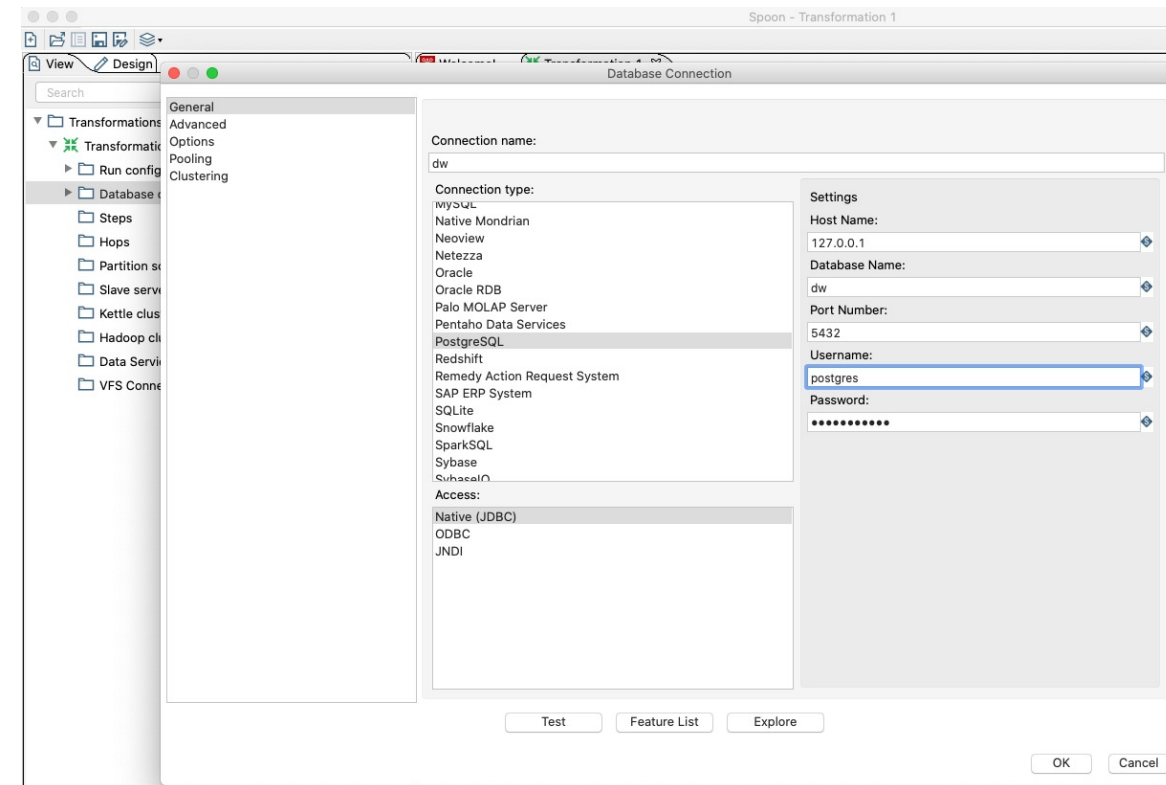
Unzip to obtain the folder: schema-workbench



# Example of ETL and Data Warehouse

# Database connection in PDI

- New transformation
- New database connection in View
  - Host server address/location: localhost  
127.0.0.1 take a look to your url  
<http://127.0.0.1:60624/>
  - Name database: dw
  - Password
  - User: postgres
- Share



# Example Sales: import Table by PDI

CSV file input

Step name: CSV file input

Filename: /Users/linda/Documents/DWH\_PDI/L5/SuperstoreSales.csv [Browse...](#)

Delimiter: ; [Insert TAB](#)

Enclosure: "

NIO buffer size: 50000

Lazy conversion? ☒

Header row present? ☒

Add filename to result ☐

The row number field name (optional)

Running in parallel? ☐

New line possible in fields? ☐

Format: mixed

File encoding:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	Row ID	Integer	#	15	0	\$	,	.	none
2	Order Code	Integer	#	15	0	\$	,	.	none
3	Order Date	Date	dd/MM/yyyy			\$	,	.	none
4	Order Priority	String		200		\$	,	.	none
5	Order Quantity	Integer	#	15	0	\$	,	.	none
6	Sales	Number	###	15	0	\$	,	.	none
7	Discount	Number	###	15	0	\$	,	.	none
8	Ship Mode	String		200		\$	,	.	none
9	Profit	Number	###	15	0	\$	,	.	none
10	Unit Price	Number	###	15	0	\$	,	.	none
11	Shipping Cost	Number	###	15	0	\$	,	.	none
12	Customer Code	Integer	#	15	0	\$	,	.	none
13	Customer Name	String		200		\$	,	.	none
14	Province	String		200		\$	,	.	none
15	Region	String		200		\$	,	.	none
16	Customer Segment	String		200		\$	,	.	none
17	Product Code	Integer	#	15	0	\$	,	.	none
18	Product Category	String		200		\$	,	.	none
19	Product Sub-Category	String		200		\$	,	.	none
20	Product Name	String		200		\$	,	.	none
21	Product Container	String		200		\$	,	.	none
22	Product Base Margin	Number	###	15	0	\$	,	.	none
23	Ship Date	Date	dd/MM/yyyy			\$	,	.	none

Check Length and  
Format for Number  
type will be ###

# Example Sales: import and create Table by PDI



Table output

Step name: Table output

Connection: dw

Target schema: public

Target table: write\_test

Commit size: 1000

Truncate table: ☒

Ignore insert errors: ☐

Specify database fields: ☒

Main options Database fields

Fields to insert:

#	Table field	Stream field
1	Row ID	Row ID
2	Order Code	Order Code
3	Order Date	Order Date
4	Order Priority	Order Priority
5	Order Quantity	Order Quantity
6	Sales	Sales
7	Discount	Discount
8	Ship Mode	Ship Mode
9	Profit	Profit
10	Unit Price	Unit Price
11	Shipping Cost	Shipping Cost
12	Customer Code	Customer Code
13	Customer Name	Customer Name
14	Province	Province
15	Region	Region

Get fields

Enter field mapping

Help OK Cancel SQL

Simple SQL editor

SQL statements, separated by semicolon ';'

```
CREATE TABLE "public".write_test
(
  "Row ID" BIGINT
  , "Order Code" BIGINT
  , "Order Date" TIMESTAMP
  , "Order Priority" VARCHAR(200)
  , "Order Quantity" BIGINT
  , Sales BIGINT
  , Discount BIGINT
  , "Ship Mode" VARCHAR(200)
  , Profit BIGINT
  , "Unit Price" BIGINT
  , "Shipping Cost" BIGINT
  , "Customer Code" BIGINT
  , "Customer Name" VARCHAR(200)
  , Province VARCHAR(200)
  , Region VARCHAR(200)
  , "Customer Segment" VARCHAR(200)
  , "Product Code" BIGINT
  , "Product Category" VARCHAR(200)
  , "Product Sub-Category" VARCHAR(200)
  , "Product Name" VARCHAR(200)
  , "Product Container" VARCHAR(200)
  , "Product Base Margin" BIGINT
  , "Ship Date" TIMESTAMP
)
;
```

Line 1 column 0

Execute Clear cache Close

Help OK Cancel SQL

createTableSales.ktr

# Example Sales: the table in the DW

127.0.0.1:60624/browser/

pgAdmin File Object Tools Help

Browser Dashboard Properties SQL Statistics Dependencies Dependents public.write\_test/dw/postgres@PostgreSQL 10

public.write\_test/dw/postgres@PostgreSQL 10

Query Editor Query History

```
1 SELECT * FROM public.write_test
2
```

Data Output Explain Messages Notifications

Row ID	Order Code	Order Date	Order Priority	Order Quantity	sales	discount	Ship Mode
bigint	bigint	timestamp without time zone	character varying (200)	bigint	bigint	bigint	character varying (200)
1	1	2010-10-13 00:00:00	Low	6	262	0	Regular Air
2	49	2012-10-01 00:00:00	High	49	10123	0	Delivery Truck
3	50	2012-10-01 00:00:00	High	27	245	0	Regular Air
4	80	2011-07-10 00:00:00	High	30	4966	0	Regular Air
5	85	2010-08-28 00:00:00	Not Specified	19	394	0	Regular Air
6	86	2010-08-28 00:00:00	Not Specified	21	147	0	Regular Air
7	97	2011-06-17 00:00:00	High	12	94	0	Regular Air
8	98	2011-06-17 00:00:00	High	22	905	0	Regular Air
9	103	2011-03-24 00:00:00	High	21	2782	0	Express Air
10	107	2010-02-26 00:00:00	Low	44	228	0	Regular Air
11	127	2010-11-23 00:00:00	Medium	45	197	0	Regular Air
12	128	2010-11-23 00:00:00	Medium	32	125	0	Regular Air
13	134	2012-06-08 00:00:00	Not Specified	32	717	0	Regular Air
14	135	2012-06-08 00:00:00	Not Specified	31	1474	0	Regular Air
15	149	2012-08-04 00:00:00	Not Specified	15	81	0	Regular Air
16	160	2011-05-30 00:00:00	Medium	46	1815	0	Regular Air
17	161	2009-11-25 00:00:00	Not Specified	16	248	0	Regular Air
18	175	2012-02-14 00:00:00	Critical	44	4462	0	Delivery Truck
19	176	2012-02-14 00:00:00	Critical	11	664	0	Regular Air
20	203	2012-04-15 00:00:00	Low	15	835	0	Regular Air
21	204	2012-04-15 00:00:00	Low	18	2481	0	Regular Air
22	213	2010-03-12 00:00:00	Not Specified	13	59	0	Express Air
23	214	2010-03-12 00:00:00	Not Specified	21	97	0	Regular Air
24	229	2011-03-09 00:00:00	Low	33	512	0	Regular Air
25	230	2011-03-09 00:00:00	Low	38	185	0	Regular Air
26	231	2012-08-04 00:00:00	High	30	81	0	Regular Air
27	249	2011-05-06 00:00:00	High	23	67	0	Regular Air
28	250	2010-12-23 00:00:00	High	25	12028	0	Delivery Truck

# Example create a job – create staging tables

etl\_process.kjb



orders\_stg.ktr      products\_stg.ktr      customers\_stg.ktr

SQL --> Execute  
in the transformations to create the tables

# Example – orders staging table

orders\_stg.ktr

CSV file input

Step name: CSV file input

Filename: /Users/linda/Documents/DWH\_PDI/L5/SuperstoreSales.csv

Delimiter: ;

Enclosure: "

NIO buffer size: 50000

Lazy conversion? ☒

Header row present? ☒

Add filename to result ☐

The row number field name (optional):

Running in parallel? ☐

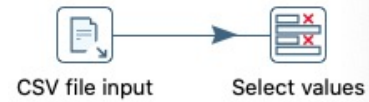
New line possible in fields? ☐

Format: mixed

File encoding:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	Row ID	Integer	#	15	0	\$	,	.	none
2	Order Code	Integer	#	15	0	\$	,	.	none
3	Order Date	Date	dd/MM/yyyy			\$	,	.	none
4	Order Priority	String		50		\$	,	.	none
5	Order Quantity	Integer	#	50	0	\$	,	.	none
6	Sales	Number	###	15	2	\$	,	.	none
7	Discount	Number	###	15	2	\$	,	.	none
8	Ship Mode	String		15		\$	,	.	none
9	Profit	Number	###	15	2	\$	,	.	none
10	Unit Price	Number	###	15	2	\$	,	.	none
11	Shipping Cost	Number	###	15	2	\$	,	.	none
12	Customer Code	Integer	#	15	0	\$	,	.	none
13	Customer Name	String		150		\$	,	.	none
14	Province	String		150		\$	,	.	none
15	Region	String		150		\$	,	.	none
16	Customer Segment	String		150		\$	,	.	none
17	Product Code	Integer	#	15	0	\$	,	.	none
18	Product Category	String		150		\$	,	.	none
19	Product Sub-Categ...	String		150		\$	,	.	none
20	Product Name	String		250		\$	,	.	none
21	Product Container	String		150		\$	,	.	none
22	Product Base Margin	Number	###	15	2	\$	,	.	none
23	Ship Date	Date	dd/MM/yyyy			\$	,	.	none

# Example – orders staging table



Select values

Step name: Select values

Select & Alter Remove Meta-data

Fields :

#	Fieldname	Rename to	Length	Precision
1	Order Code			
2	Order Priority			

Get fields to select

Edit Mapping

Include unspecified fields, ordered by ☐

Help OK Cancel

orders\_stg.ktr



# Example – orders staging table

orders\_stg.ktr



The screenshot shows the 'Sort rows' dialog box with the following configuration:

- Step name: Sort rows
- Sort directory: %%java.io.tmpdir%%
- TMP-file prefix: out
- Sort size (rows in memory): 1000000
- Free memory threshold (in %):
- Compress TMP Files?: ☐
- Only pass unique rows? (verifies ke): ☒

Fields :

#	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Collator Streng
1	Order Code	Y			

Buttons: ? Help, OK, Cancel, Get Fields

Make order code unique

# Example – orders staging table

orders\_stg.ktr

Create the table orders dimension

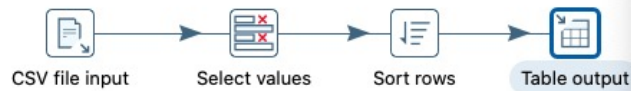


Table output

Step name: Table output

Connection: dw

Target schema: public

Target table: stg\_orders

Commit size: 1000

Truncate table: ☐

Ignore insert errors: ☐

Specify database fields: ☒

Main options | Database fields

Fields to insert:

#	Table field	Stream field
1	Order C...	Order Code
2	Order Pri...	Order Prio...

Simple SQL editor

SQL statements, separated by semicolon ';'

```
CREATE TABLE "public".stg_orders
(
  "Order Code" BIGINT
, "Order Priority" VARCHAR(50)
)
```

Line 1 column 0

Execute Clear cache Close

Get fields field mapping

Help OK Cancel SQL

Remember:  
SQL --> Execute  
to create the  
table!

# Example – products and customers staging tables

- Similar processes to the previous for the creation of the orders staging table

products\_stg.ktr

customers\_stg.ktr

Remember:  
SQL --> Execute to create  
the table!

# Example create a job – creating staging tables

dw

- Cast
- Catalogs
- Event Triggers
- Extensions
- Foreign Data Wrappers
- Languages
- Schemas (1)
  - public
    - Collations
    - Domains
    - FTS Configurations
    - FTS Dictionaries
    - FTS Parsers
    - FTS Templates
    - Foreign Tables
    - Functions
    - Materialized Views
    - Sequences
    - Tables (4)
      - stg\_customers
      - stg\_orders
      - stg\_products
      - write\_test
        - Columns
        - Constraints
        - Indexes
        - RLS Policies
        - Rules
        - Triggers

Query EditorQuery History

1SELECT \* FROM public.stg\_customers

2

Data OutputExplainMessagesNotifications

	Customer Code bigint	Customer Name character varying (150)	province character varying (150)	region character varying (150)	Customer Segment character varying (150)
1		1 Muhammed MacIntyre	Nunavut	Nunavut	Small Business
2		2 Barry French	Nunavut	Nunavut	Consumer
3		3 Clay Rozendal	Nunavut	Nunavut	Corporate
4		4 Carlos Soltero	Nunavut	Nunavut	Consumer
5		5 Carl Jackson	Nunavut	Nunavut	Corporate
6		6 Monica Federle	Nunavut	Nunavut	Corporate
7		7 Dorothy Badders	Nunavut	Nunavut	Home Office
8		8 Neola Schneider	Nunavut	Nunavut	Home Office
9		9 Carlos Daly	Nunavut	Nunavut	Home Office
10		10 Claudia Miner	Nunavut	Nunavut	Small Business
11		11 Allen Rosenblatt	Nunavut	Nunavut	Small Business
12		12 Sylvia Foulston	Nunavut	Nunavut	Home Office
13		13 Jim Radford	Nunavut	Nunavut	Corporate
14		14 Carl Ludwig	Nunavut	Nunavut	Corporate
15		15 Don Miller	Nunavut	Nunavut	Home Office
16		16 Annie Cyprus	Nunavut	Nunavut	Home Office
17		17 Grant Carroll	Nunavut	Nunavut	Small Business
18		18 Alan Barnes	Nunavut	Nunavut	Corporate
19		19 Jack Garza	Nunavut	Nunavut	Corporate
20		20 Julia West	Nunavut	Nunavut	Corporate
21		21 Eugene Barchas	Nunavut	Nunavut	Corporate
22		22 Edward Hooks	Nunavut	Nunavut	Consumer

createTableSalesetl\_processorders\_stgproducts\_stgcustomers\_stg

100%

StartTransformation - orders stgTransformation - products stgTransformation - customers stgSuccess

Query EditorQuery History

1SELECT \* FROM public.stg\_orders

2

Data OutputExplainMessagesNotifications

	Order Code bigint	Order Priority character varying
1	3	Low
2	6	Not Specified
3	32	High
4	35	Not Specified
5	36	Critical
6	65	Critical
7	66	Low
8	69	Not Specified
9	70	Low
10	96	High
11	97	Medium
12	129	Low
13	130	High
14	132	Medium
15	134	Not Specified
16	135	Not Specified
17	166	High
18	193	Critical
19	194	Medium
20	195	Medium
21	197	High
22	224	Not Specified
23	225	Critical

Query EditorQuery History

1SELECT \* FROM public.stg\_products

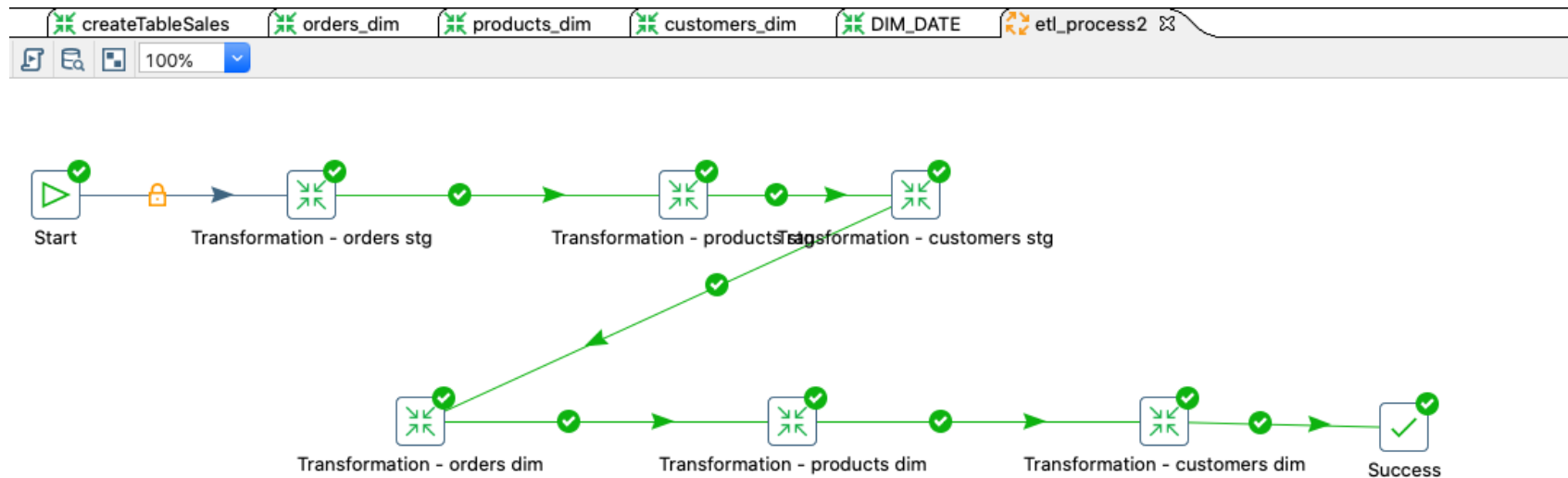
2

Data OutputExplainMessagesNotifications

	Product Code bigint	Product Category character varying (150)	Product Sub-Category character varying (150)	Product Name character varying (250)	Product Container character varying (150)
1	1	Office Supplies	Storage & Organization	Eldon Base for stackable stor...	Large Box
2	2	Office Supplies	Appliances	1.7 Cubic Foot Compact 'Cub...	Jumbo Drum
3	3	Office Supplies	Binders and Binder Accessories	Cardinal Slant-D Ring Binder...	Small Box
4	4	Technology	Telephones and Communicati...	R380	Small Box
5	5	Office Supplies	Appliances	Holmes HEPA Air Purifier	Medium Box
6	6	Furniture	Office Furnishings	G.E. Longer-Life Indoor Reces...	Small Pack
7	7	Office Supplies	Binders and Binder Accessories	Angle-D Binders with Locking ...	Small Box
8	8	Office Supplies	Storage & Organization	SAFCO Mobile Desk Side File, ...	Small Box
9	9	Office Supplies	Storage & Organization	SAFCO Commercial Wire Shel...	Large Box
10	10	Office Supplies	Paper	Xerox 198	Small Box
11	11	Office Supplies	Paper	Xerox 1980	Small Box
12	12	Office Supplies	Rubber Bands	Advantus Map Pennant Flags ...	Wrap Bag
13	13	Technology	Computer Peripherals	DS/HD IBM Formatted Diskett...	Small Pack
14	14	Office Supplies	Binders and Binder Accessories	Wilson Jones 1" Hanging Dubl...	Small Box
15	15	Furniture	Office Furnishings	Ultra Commercial Grade Dual ...	Wrap Bag
16	16	Office Supplies	Envelopes	#10-4 1/8" x 9 1/2" Premium ...	Small Box
17	17	Furniture	Bookcases	Hon 4-Shelf Metal Bookcases	Jumbo Box
18	18	Furniture	Tables	Lesro Sheffield Collection Cof...	Large Box
19	19	Technology	Telephones and Communicati...	g520	Small Box
20	20	Technology	Telephones and Communicati...	LX 788	Small Box
21	21	Office Supplies	Labels	Avery 52	Small Box
22	22	Office Supplies	Rubber Bands	Plymouth Boxed Rubber Band...	Wrap Bag
23	23	Office Supplies	Binders and Binder Accessories	GBC Pre-Punched Binding Pa...	Small Box
24	24	Technology	Computer Peripherals	Maxell 3.5" DS/HD IBM-Forma...	Small Pack
25	25	Office Supplies	Pens & Art Supplies	Newell 335	Wrap Bag
26	26	Office Supplies	Pens & Art Supplies	SANFORD Liquid Accent Dia...	Wrap Bag

# Example create a job – creating dimension tables

etl\_process2.kjb



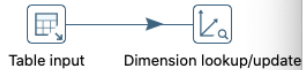
orders\_dim.ktr

products\_dim.ktr

customers\_dim.ktr

SQL --> Execute  
in the transformations to create the tables

# Example – orders dimension



Dimension lookup/update

Step name: Dimension lookup/update

Update the dimension? ☒

Connection: dw Edit... New... Wizard... Browse...

Target schema: Browse...

Target table: dim\_orders Browse...

Commit size: 100

Enable the cache? ☒

Pre-load the cache? ☐

Cache size in rows (0 = cache all): 5000

Keys Fields

#	Dimension field	Field in stream
1	Order Code	Order Code

Technical key field: surrkey order New name:

Creation of technical key

☒ Use table maximum + 1

☐ Use sequence

☐ Use auto increment field

Version field: version

Stream Datefield:

Date range start field: date from Min. year: 1900

Use an alternative start date? ☐ <Select Option>

Table date range end: date to Max. year: 2199

OK Cancel Get Fields SQL

Help

Dimension lookup/update

Step name: Dimension lookup/update

Update the dimension? ☒

Connection: dw Edit... New... Wizard... Browse...

Target schema: Browse...

Target table: dim\_orders Browse...

Commit size: 100

Enable the cache? ☒

Pre-load the cache? ☐

Cache size in rows (0 = cache all): 5000

Keys Fields

#	Dimension field	Stream field to compare with	Type of dimension update
1	Order Priority	Order Priority	Update

Technical key field: surrkey order New name:

Creation of technical key

☒ Use table maximum + 1

☐ Use sequence

☐ Use auto increment field

Version field: version

Stream Datefield:

Date range start field: date from Min. year: 1900

Use an alternative start date? ☐ <Select Option>

Table date range end: date to Max. year: 2199

OK Cancel Get Fields SQL

Help

Dimension lookup/update

Step name: Dimension lookup/update

Update the dimension? ☒

Connection: dw Edit... New... Wizard... Browse...

Target schema: Browse...

Target table: dim\_orders Browse...

Commit size: 100

Enable the cache? ☒

Pre-load the cache? ☐

Cache size in rows (0 = cache all): 5000

Keys Fields

#	Dimension field	Field in stream
1	Order Code	Order Co

Technical key field:

Version field: version

Stream Datefield:

Date range start field: date from Min. year: 1900

Use an alternative start date? ☐ <Select Option>

Table date range end: date to Max. year: 2199

OK Cancel Get Fields SQL

Help

Simple SQL editor

SQL statements, separated by semicolon ';'

```
CREATE TABLE dim_orders
(
  surrkey_order BIGSERIAL
, version INTEGER
, date_from TIMESTAMP
, date_to TIMESTAMP
, "Order Code" BIGINT
, "Order Priority" VARCHAR(50)
)
;CREATE INDEX idx_dim_orders_lookup ON dim_order
```

Line 1 column 0

Execute Clear cache Close

Remember:  
SQL -->  
Execute to  
create the  
table!

# Example – products and customers dimension

- Similar processes as for the orders dimension tables

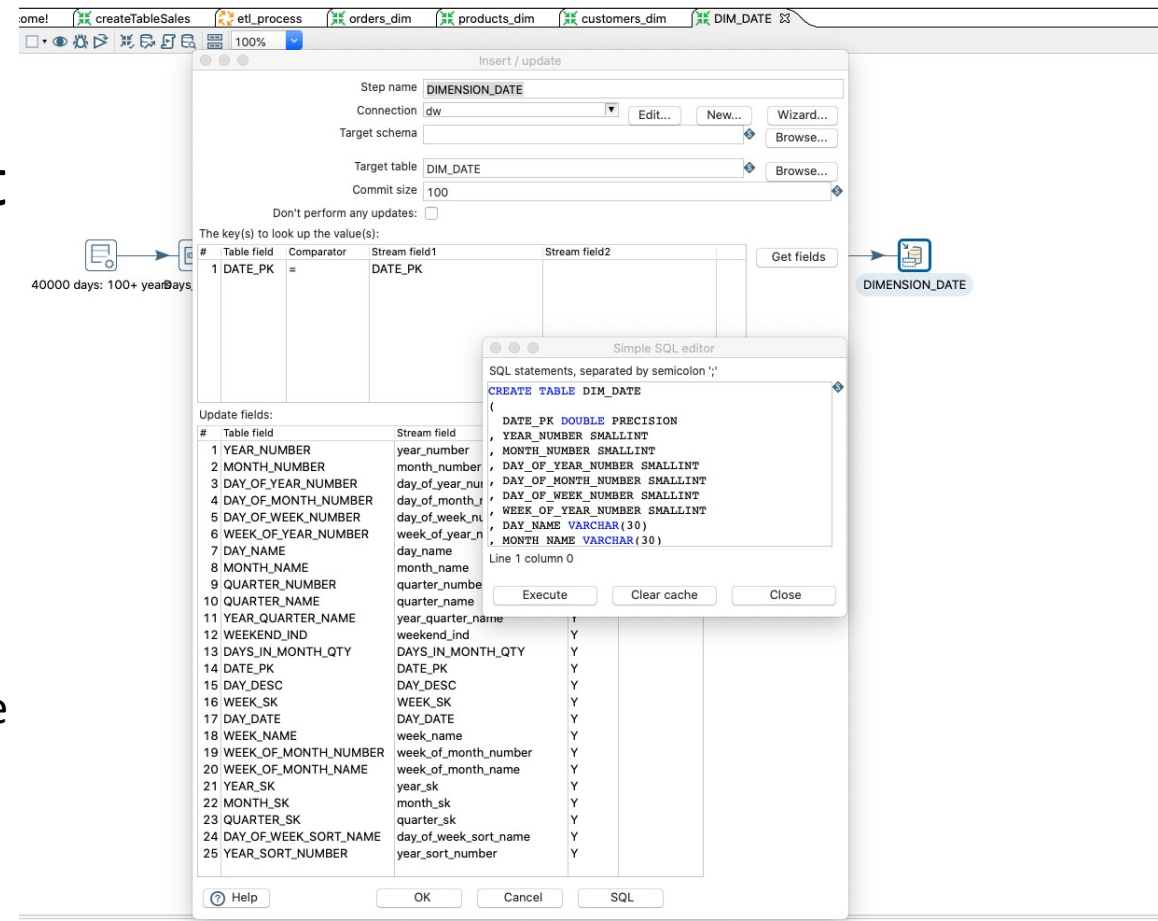
Remember:  
SQL --> Execute to create  
the table!

# Example – time dimension

- In addition we use a dim\_date transformation
- The time doesn't change so we will not execute this function anymore, it is only execute one time

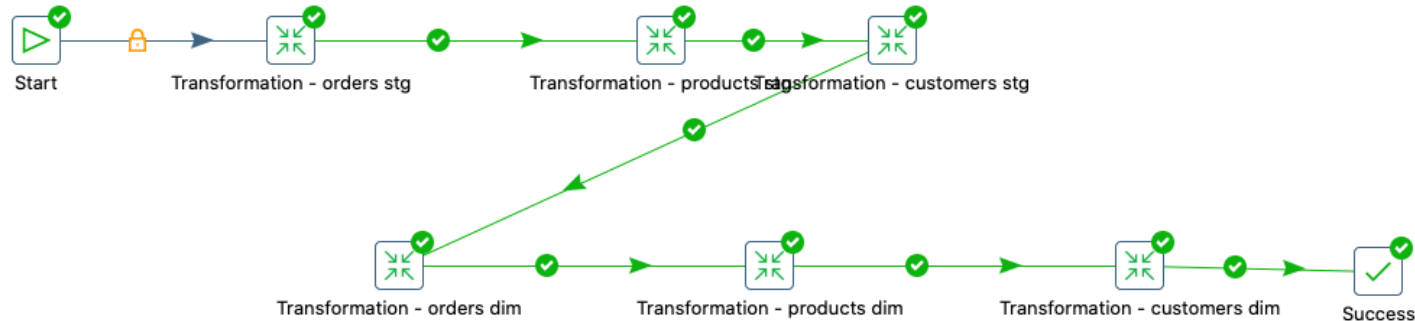
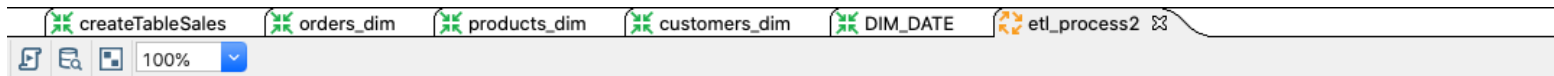
DIM\_DATE.ktr

Remember:  
SQL --> Execute to create  
the table!





# Example create a job – creating dimension tables



+ DIM\_DATE.ktr

- ▼ Tables (8)
  - > dim\_customers
  - > dim\_date
  - > dim\_orders
  - > dim\_products
  - > stg\_customers
  - > stg\_orders
  - > stg\_products
  - ▼ write\_test

# Example – creating the staging quantitative values table

quantvalues\_stg.ktr



Select values

Step name: Select values

Select & Alter Remove Meta-data

Fields :

#	Fieldname	Rename to	Length	Precision
1	Order Code			
2	Order Date			
3	Order Quantity			
4	Sales			
5	Discount			
6	Profit			
7	Unit Price			
8	Shipping Cost			
9	Customer Code			
10	Product Code			
11	Product Base Margin			

Get fields to select

Edit Mapping

Include unspecified fields, ordered by nar ☐

Help OK Cancel

We keep only  
quantitative values  
and id for dimensions

# Example – creating the staging quantitative values table

quantvalues\_stg.ktr



Table output

Step name: Table output

Connection: dw

Target schema: public

Target table: stg\_quantvalues

Commit size: 1000

Truncate table: ☐

Ignore insert errors: ☐

Specify database fields: ☒

Main options | Database fields

Fields to insert:

#	Table field	Stream field
1	Order Code	Order Code
2	Order Date	Order Date
3	Order Quantity	Order Quantity
4	Sales	Sales
5	Discount	Discount
6	Profit	Profit
7	Unit Price	Unit Price
8	Shipping Cost	Shipping Cost
9	Customer Code	Customer Code
10	Product Code	Product Code
11	Product Base Margin	Product Base Margin

Get fields

Simple SQL editor

SQL statements, separated by semicolon ';'

```
CREATE TABLE "public".stg_quantvalues
(
  "Order Code" BIGINT
, "Order Date" TIMESTAMP
, "Order Quantity" NUMERIC(50, 0)
, Sales NUMERIC(17, 2)
, Discount NUMERIC(17, 2)
, Profit NUMERIC(17, 2)
, "Unit Price" NUMERIC(17, 2)
, "Shipping Cost" NUMERIC(17, 2)
, "Customer Code" BIGINT
)
```

Line 1 column 0

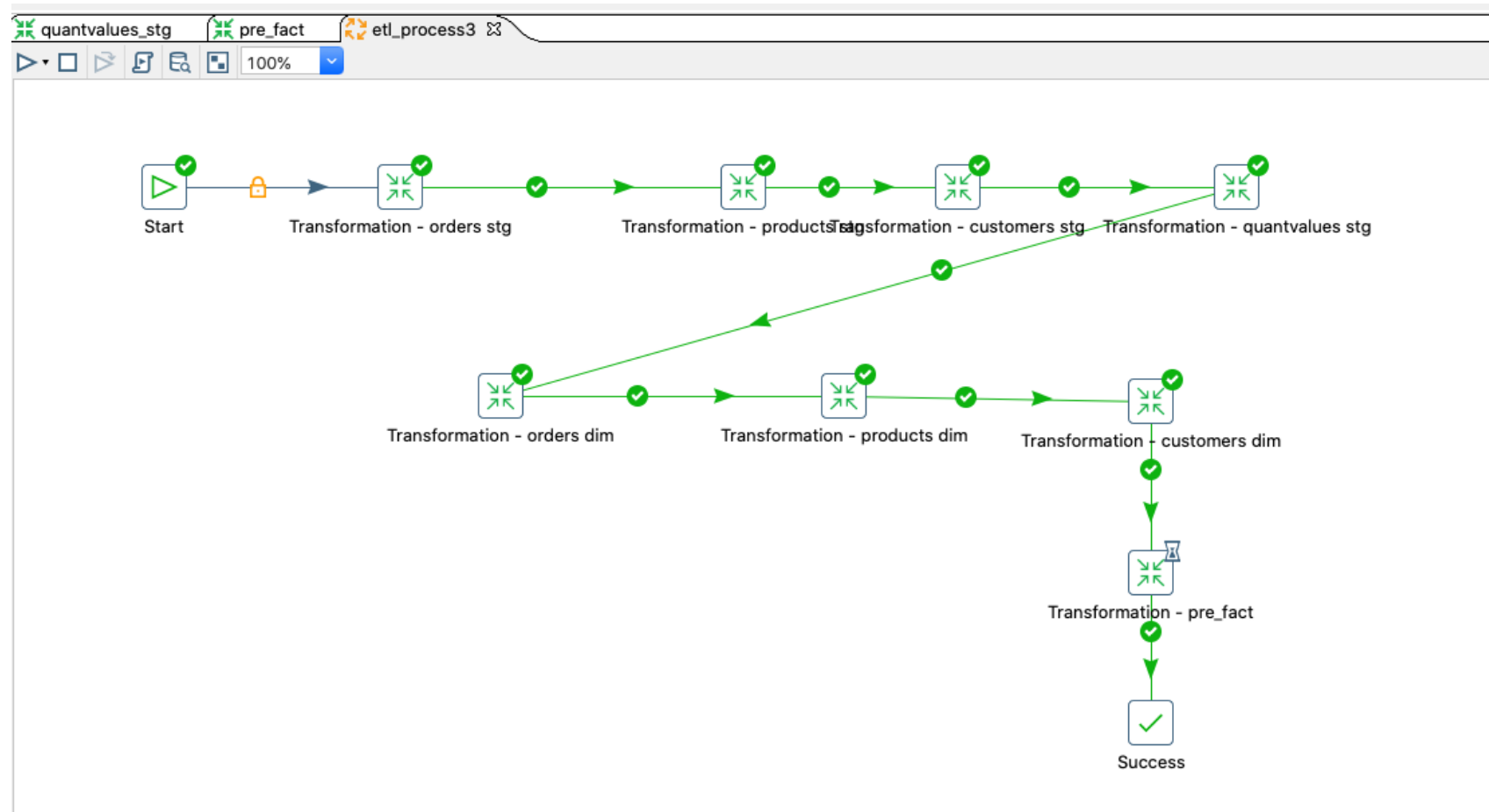
Execute Clear cache Close

We keep only quantitative values and id for dimensions

Remember:  
SQL --> Execute to create the table!

# Example create a job – creating a pre-fact table

etl\_process3.kjb



pre\_fact.ktr

Remember:  
SQL --> Execute to create  
the table!

# Example – creating a pre-fact table




Table input

Step name: Table input

Connection: dw

Buttons: Edit..., New..., Wizard..., Get SQL select statement...

SQL

```
SELECT
  "Order Code"
, "Order Date"
, "Order Quantity"
, sales
, discount
, profit
, "Unit Price"
, "Shipping Cost"
, "Customer Code"
, "Product Code"
, "Product Base Margin"
FROM "public".stg_quantvalues
```

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 0

Buttons: ? Help, OK, Preview, Cancel

pre\_fact.ktr

# Example – creating a pre-fact table

Diagram showing the flow from **Table input** to **Database lookup**.

**Database lookup**

Step name: Database lookup - orders

Connection: dw

Lookup schema: public

Lookup table: dim\_orders

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	Order Code	=	Order Code	
2	date_from	<	Order Date	
3	date_to	>=	Order Date	

Values to return from the lookup table :

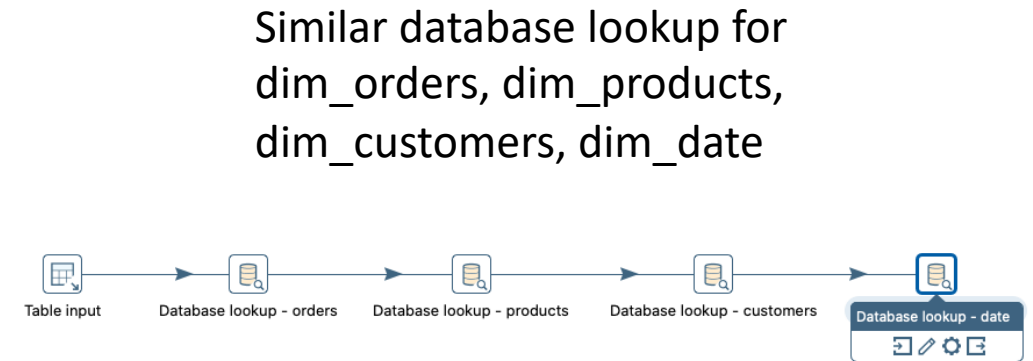
#	Field	New name	Default	Type
1	surrkey_order	foreignkey_order		Integer
2				

Do not pass the row if the lookup fails ☐

Fail on multiple results? ☐

Order by:

Buttons: Help, OK, Cancel, Get Fields, Get lookup fields



pre\_fact.ktr

# Example – creating a pre-fact table

Select only the foreign keys and the quantitative values in Select values

The screenshot displays the data integration tool interface. At the top, a workflow diagram shows the sequence: Table input → Database lookup - orders → Database lookup - products → Database lookup - customers → Database lookup - date. Below this, the 'Table output' configuration window is open, showing the step name 'Table output - pre\_fact', connection 'dw', target schema 'public', and target table 'pre\_fact'. The 'Database fields' tab is selected, showing a list of fields to insert with their corresponding stream fields. The 'Simple SQL editor' window is also open, displaying the SQL statement to create the 'pre\_fact' table with various columns and data types.

Step name: Table output - pre\_fact  
Connection: dw  
Target schema: public  
Target table: pre\_fact  
Commit size: 1000  
Truncate table: ☐  
Ignore insert errors: ☐  
Specify database fields: ☒

Fields to insert:

#	Table field	Stream field
1	Order Quantity	Order Quantity
2	sales	sales
3	discount	discount
4	profit	profit
5	Unit Price	Unit Price
6	Shipping Cost	Shipping Cost
7	Product Base Margin	Product Base Margin
8	foreignkey_order	foreignkey_order
9	foreignkey_product	foreignkey_product
10	foreignkey_customers	foreignkey_customers
11	foreignkey_date	foreignkey_date

SQL statements, separated by semicolon ';'

```
CREATE TABLE "public".pre_fact  
(  
  "Order Quantity" NUMERIC(50, 0)  
, sales NUMERIC(19, 2)  
, discount NUMERIC(19, 2)  
, profit NUMERIC(19, 2)  
, "Unit Price" NUMERIC(19, 2)  
, "Shipping Cost" NUMERIC(19, 2)  
, "Product Base Margin" NUMERIC(19, 2)  
, foreignkey_order BIGINT  
, foreignkey_product BIGINT  
)
```

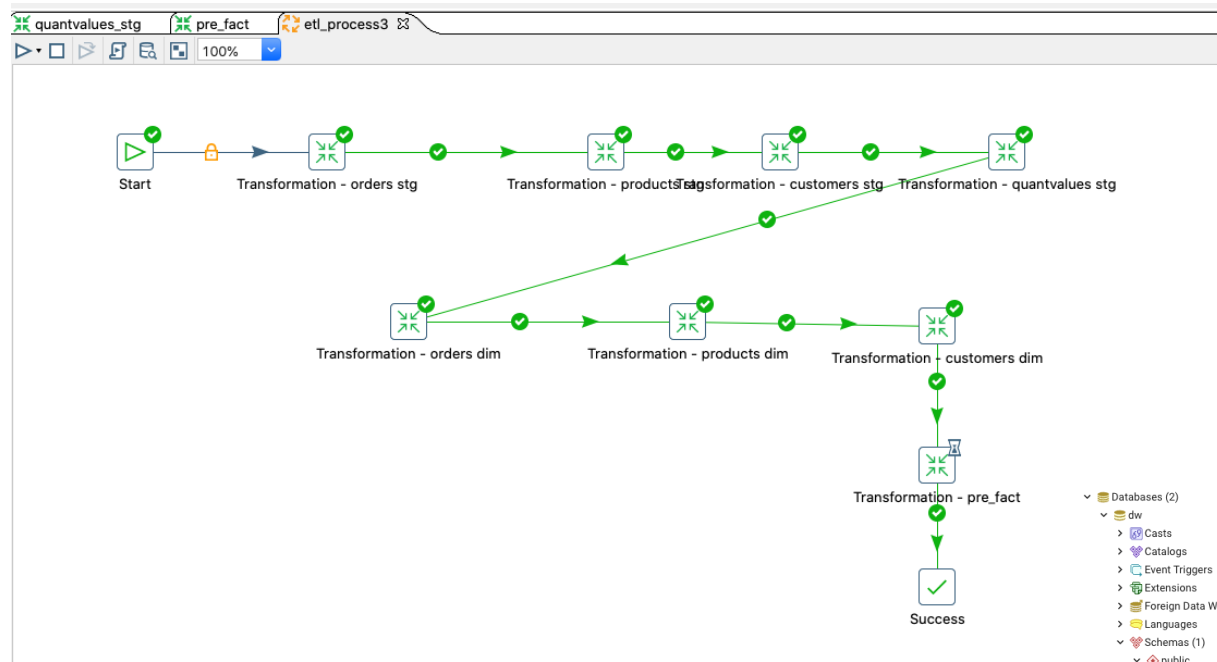
pre\_fact.ktr

The 'Select values' dialog box is shown, with the 'Step name' set to 'Select values'. The 'Meta-data' tab is selected, displaying a table of fields with columns for 'dname', 'Rename to', 'Length', and 'Precision'. The fields listed include 'Order Code', 'Order Date', 'Order Quantity', 'Sales', 'Unit Price', 'Shipping Cost', 'Product Code', 'Product Base Margin', 'foreignkey\_order', 'foreignkey\_product', 'foreignkey\_customers', and 'foreignkey\_date'. The 'Get fields to select' button is visible.

dname	Rename to	Length	Precision
Order Code			
Order Date			
Order Quantity			
Sales			
Unit Price			
Shipping Cost			
Product Code			
Product Base Margin			
foreignkey_order			
foreignkey_product			
foreignkey_customers			
foreignkey_date			

Remember:  
SQL --> Execute to create  
the table!

# Example create a job – creating pre-fact table



etl\_process3.kjb

postgres@postgres-10

Query Editor

Query History

1 SELECT \* FROM public.pre\_fact

2

Scratch Pad

Data Output

Expand

Messages

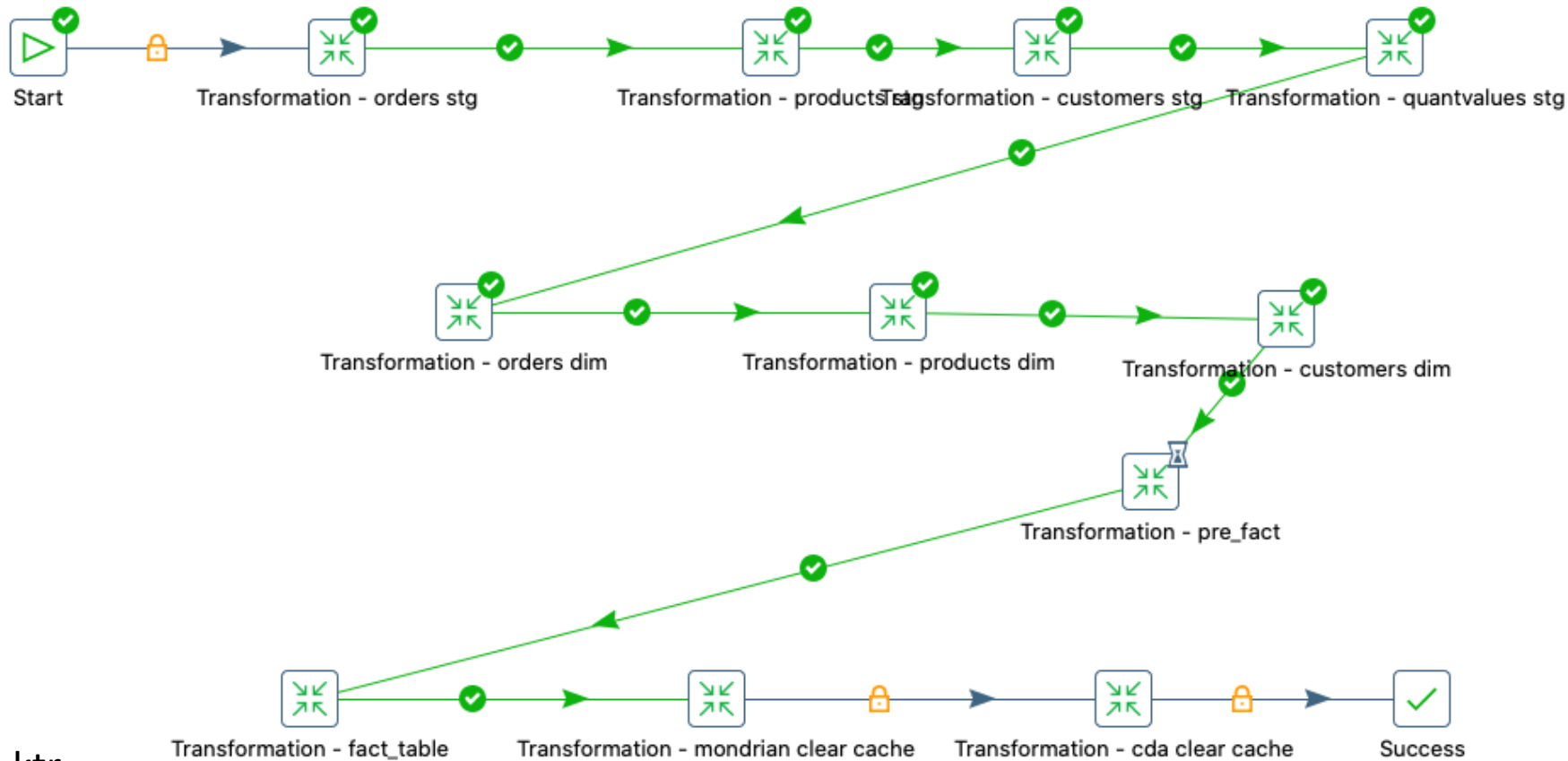
Notifications

	Order Quantity numeric (50)	sales numeric (19,2)	discount numeric (19,2)	profit numeric (19,2)	Unit Price numeric (19,2)	Shipping Cost numeric (19,2)	Product Base Margin numeric (19,2)	foreignkey_order bigint	foreignkey_product bigint	foreignkey_ bigint
1	6	261.54	0.04	-213.25	38.94	35.00	0.80	1	1	
2	49	10123.02	0.07	457.81	208.16	68.02	0.58	32	2	
3	27	244.57	0.01	46.71	8.69	2.99	0.39	32	3	
4	30	4965.76	0.08	1198.97	195.99	3.99	0.58	54	4	
5	19	394.27	0.08	30.94	21.78	5.94	0.50	58	5	
6	21	146.69	0.05	4.43	6.64	4.95	0.37	58	6	
7	12	93.54	0.03	-54.04	7.30	7.72	0.38	64	7	
8	22	905.08	0.09	127.70	42.76	6.22	[null]	64	8	
9	21	2781.82	0.07	-695.26	138.14	35.00	[null]	67	9	
10	44	228.41	0.07	-226.36	4.98	8.33	0.38	71	10	
11	45	196.85	0.01	-166.85	4.28	6.18	0.40	85	11	
12	32	124.56	0.04	-14.33	3.95	2.00	0.53	85	12	
13	32	716.84	0.00	134.72	21.78	5.94	0.50	91	5	
14	31	1474.33	0.04	114.46	47.98	3.61	0.71	91	13	
15	15	80.61	0.02	-4.72	5.28	2.99	0.37	101	14	
16	46	1815.49	0.03	782.91	39.89	3.04	0.53	108	15	
17	16	248.26	0.07	93.80	15.74	1.39	0.40	109	16	
18	44	4462.23	0.04	440.72	100.98	26.22	0.60	121	17	
19	11	663.78	0.25	-481.04	71.37	69.00	0.68	121	18	
20	15	834.90	0.06	-11.68	65.99	5.26	0.59	139	19	
21	18	2480.92	0.01	313.58	155.99	8.99	0.58	139	20	
22	13	59.03	0.10	26.92	3.69	0.50	0.38	145	21	



# Example create a job – creating a fact table

etl\_process4.kjb



fact\_table.ktr

mondrian\_clear\_cache.ktr

cda\_clear\_cache.ktr

Remember:  
SQL --> Execute to create  
the table!

# Example – creating a fact table



Table output

Step name: Table output

Connection: dw [Edit... New... Wizard...]

Target schema: public [Browse...]

Target table: fact\_table [Browse...]

Commit size: 1000

Truncate table: ☐

Ignore insert errors: ☐

Specify database fields: ☐

Main options | Database fields

#	Table field	Stream field
1	Order Quantity	Order Quantity
2	sales	sales
3	discount	discount
4	profit	profit
5	Unit Price	Unit Price
6	Shipping Cost	Shipping Cost
7	Product Base Margin	Product Base Margin
8	foreignkey_order	foreignkey_order
9	foreignkey_product	foreignkey_product
10	foreignkey_customers	foreignkey_customers
11	foreignkey_date	foreignkey_date

Simple SQL editor

SQL statements, separated by semicolon ';'

```
CREATE TABLE "public".fact_table
(
  "Order Quantity" NUMERIC(50, 0)
, sales NUMERIC(21, 2)
, discount NUMERIC(21, 2)
, profit NUMERIC(21, 2)
, "Unit Price" NUMERIC(21, 2)
, "Shipping Cost" NUMERIC(21, 2)
, "Product Base Margin" NUMERIC(21, 2)
, foreignkey_order BIGINT
, foreignkey_product BIGINT
)
```

Line 1 column 1

Execute Clear cache Close

Help OK Cancel SQL

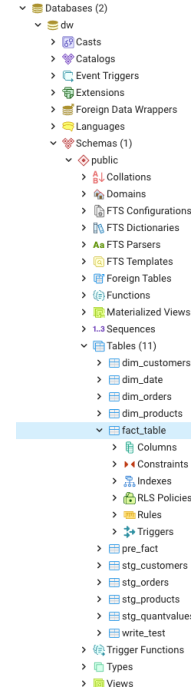
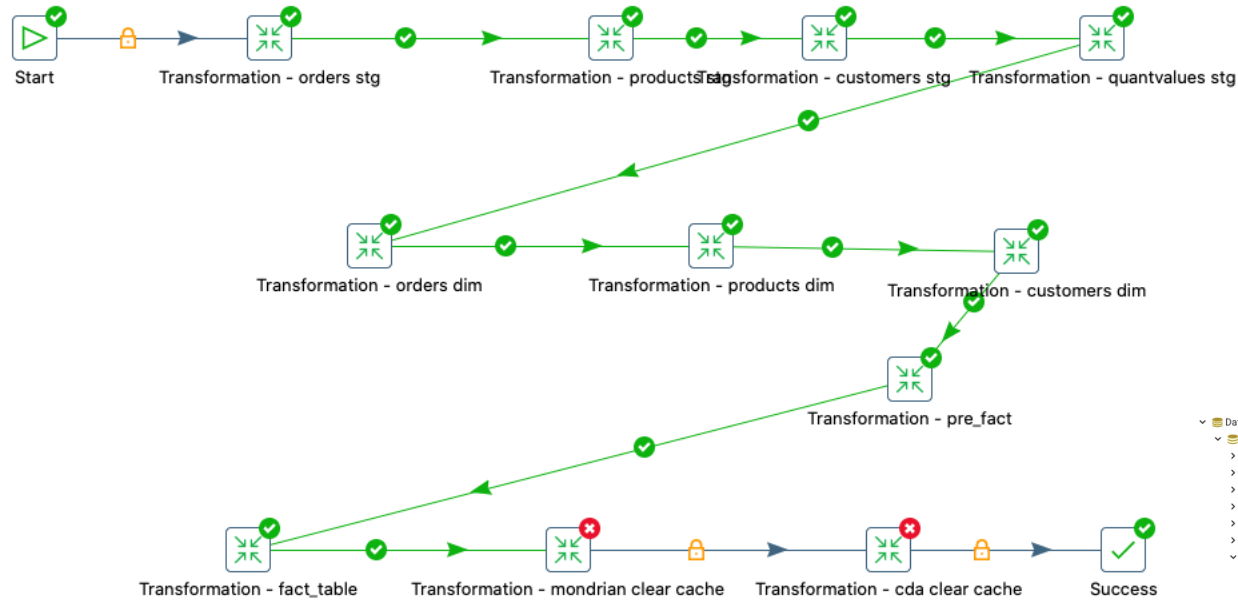
fact\_table.ktr

# Clear the cache of our system



cda\_clear\_cache.ktr

# Example create a job – creating a fact table



public.fact\_table/dw/postgres@PostgreSQL 10

Query Editor

Query History

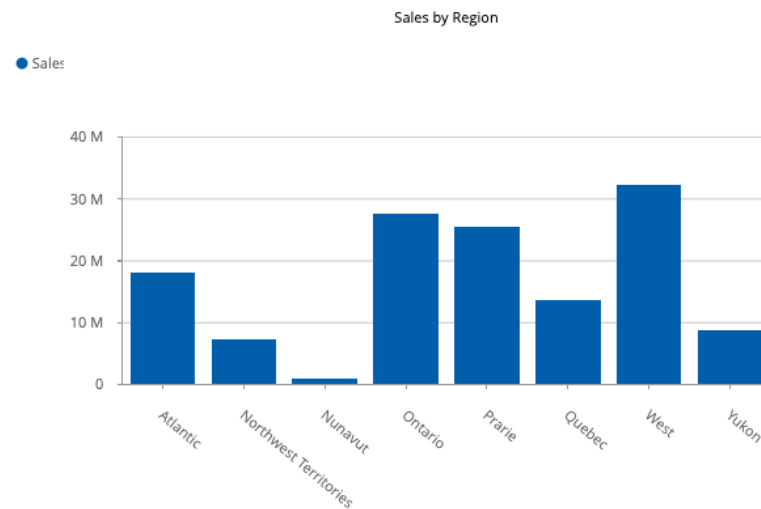
1 SELECT \* FROM public.fact\_table

2

Scratch Pad

Data	Output	Explain	Messages	Notifications						
	Order Quantity numeric (50)	sales numeric (21,2)	discount numeric (21,2)	profit numeric (21,2)	Unit Price numeric (21,2)	Shipping Cost numeric (21,2)	Product Base Margin numeric (21,2)	foreignkey_order bigint	foreignkey_product bigint	foreignkey_ bigint
1	6	261.54	0.04	-213.25	38.94	35.00	0.80	1	1	1
2	49	10123.02	0.07	457.81	208.16	68.02	0.58	32	2	2
3	27	244.57	0.01	46.71	8.69	2.99	0.39	32	3	3
4	30	4965.76	0.08	1198.97	195.99	3.99	0.58	54	4	4
5	19	394.27	0.08	30.94	21.78	5.94	0.50	58	5	5
6	21	146.69	0.05	4.43	6.64	4.95	0.37	58	6	6
7	12	93.54	0.03	-54.04	7.30	7.72	0.38	64	7	7
8	22	905.08	0.09	127.70	42.76	6.22	[null]	64	8	8
9	21	2781.82	0.07	-695.26	138.14	35.00	[null]	67	9	9
10	44	228.41	0.07	-226.36	4.98	8.33	0.38	71	10	10
11	45	196.85	0.01	-166.85	4.28	6.18	0.40	85	11	11
12	32	124.56	0.04	-14.33	3.95	2.00	0.53	85	12	12
13	32	716.84	0.00	134.72	21.78	5.94	0.50	91	5	5
14	31	1474.33	0.04	114.46	47.98	3.61	0.71	91	13	13
15	15	80.61	0.02	-4.72	5.28	2.99	0.37	101	14	14
16	46	1815.49	0.03	782.91	39.89	3.04	0.53	108	15	15
17	16	248.26	0.07	93.80	15.74	1.39	0.40	109	16	16
18	44	4462.23	0.04	440.72	100.98	26.22	0.60	121	17	17
19	11	663.78	0.25	-481.04	71.37	69.00	0.68	121	18	18
20	15	834.90	0.06	-11.68	65.99	5.26	0.59	139	19	19
21	18	2480.92	0.01	313.58	155.99	8.99	0.58	139	20	20
22	13	59.03	0.10	26.92	3.69	0.50	0.38	145	21	21
23	21	97.48	0.05	-5.77	4.71	0.70	0.80	145	22	22
24	33	511.83	0.10	-172.88	15.99	13.18	0.37	155	23	23
25	38	184.99	0.05	-144.55	4.89	4.93	0.66	155	24	24
26	30	80.90	0.09	5.76	2.88	0.70	0.56	156	25	25
27	23	67.24	0.06	4.90	2.84	0.93	0.54	168	26	26
28	25	12028.23	0.01	-547.61	449.99	49.00	0.38	169	27	27

# Mashboard example



Show 10 entries

Search:

Priority	Sales	Discount	Profit
Low	29445293.16	775.44	3401532.54
High	29400466.41	785.7	3663224.1
Medium	25762068.9	736.92	2943465.57
Not Specified	25005269.79	729.54	1967263.47
Critical	24627309.93	727.11	1720426.14

Showing 1 to 5 of 5 entries

Previous

1

Next

Show 10 entries

Search:

Products.Category	Sales	Discount	Profit
Office Machines	19518274.26	153.18	2769416.37
Telephones and Communication	17003825.28	381.06	2852564.58
Copiers and Fax	10173251.7	40.32	1506253.41