Building Sales Data mart Using Pentaho Part 2 (continued)

By Naheed Anjum Arafat

Task 2.2
stage_sales → fact_sales

Objective


stage_sales
date lookup
hour_min lookup
product lookup
fact_sales

Table input

Table input

Step name: stage_sales

Connection: postgres

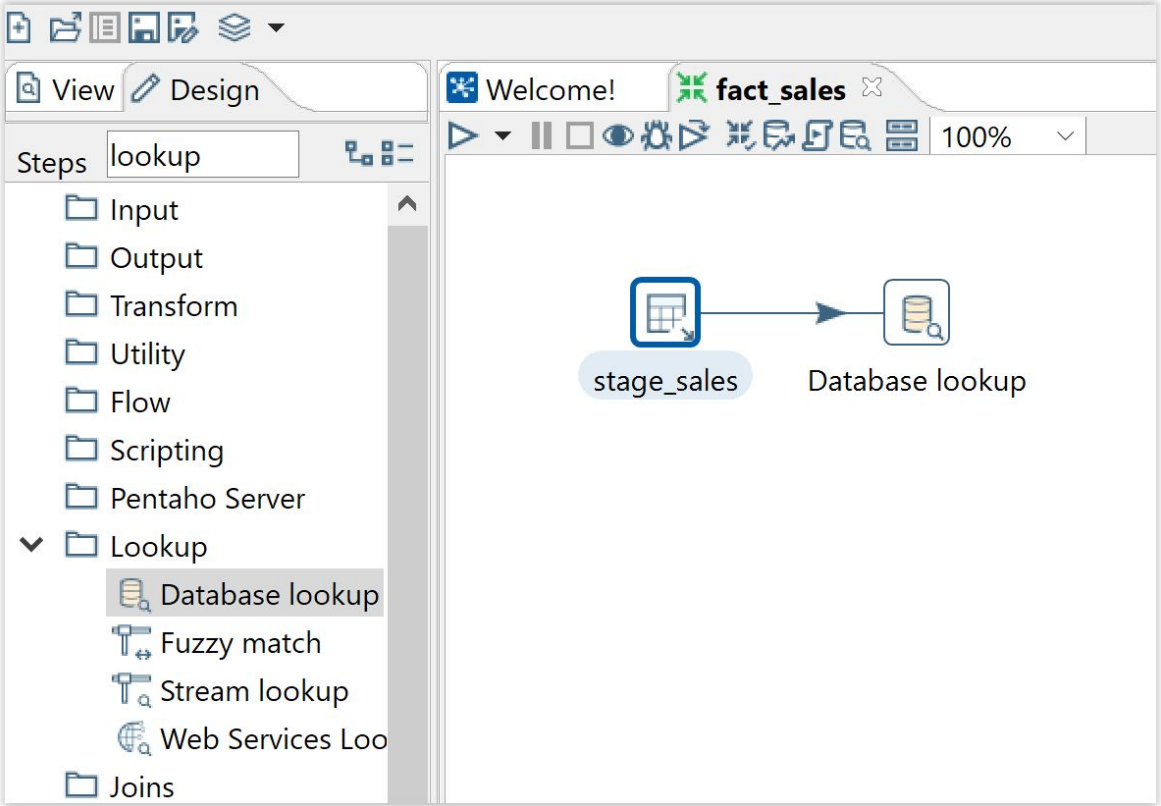
Edit... New... Wizard...

SQL

Get SQL select statement...

```
SELECT
  source_row_id
, pos_transaction_id
, sales_qty
, regular_unit_price
, extended_price
, extended_discount
, product_id
, hour_min_str
, date_str
, unit_discount
, net_unit_price
, net_sales_amount
FROM stage_sales
```

Database lookup



Database Lookup

Database Value Lookup

Step name: date lookup

Connection: postgres | Edit... | New... | Wizard...

Lookup schema: | Browse...

Lookup table: dim_date | Browse...

Enable cache?

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

Load all data from table

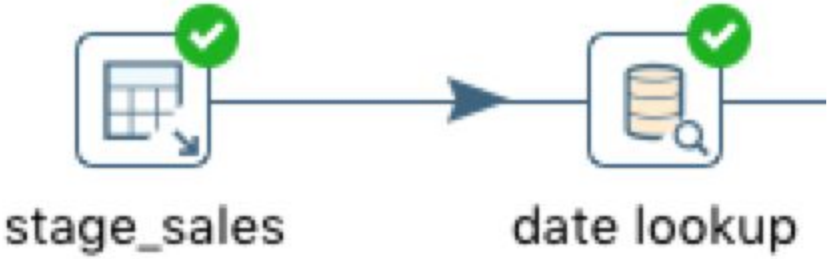
The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	date_str	=	date_str	

Values to return from the lookup table :

#	Field	New name	Default	Type
1	dim_date_key		0	Integer

dim_date table has been created for you
Make sure the data type and format is correct


stage_sales
date lookup

Database lookup dim_hour_min_key

Database Value Lookup

Step name: hour_min lookup

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

Lookup schema: [Browse...]

Lookup table: dim_hour_min [Browse...]

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	hour_min	=	hour_min_str	

Values to return from the lookup table :

#	Field	New name	Default	Type
1	dim_hour_min_key		0	Integer

Do not pass the row if the lookup fails ☐

Fail on multiple results? ☐

Order by

[Help] [OK] [Cancel] [Get Fields] [Get lookup fields]

dim_hour_min table has been created for you
Make sure the data type and format is correct


stage_sales
date lookup
hour_min lookup

Database lookup dim_product_key

Database Value Lookup

Step name: product lookup

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

Lookup schema: [Browse...]

Lookup table: dim_product [Browse...]

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	barcode	=	product_id	

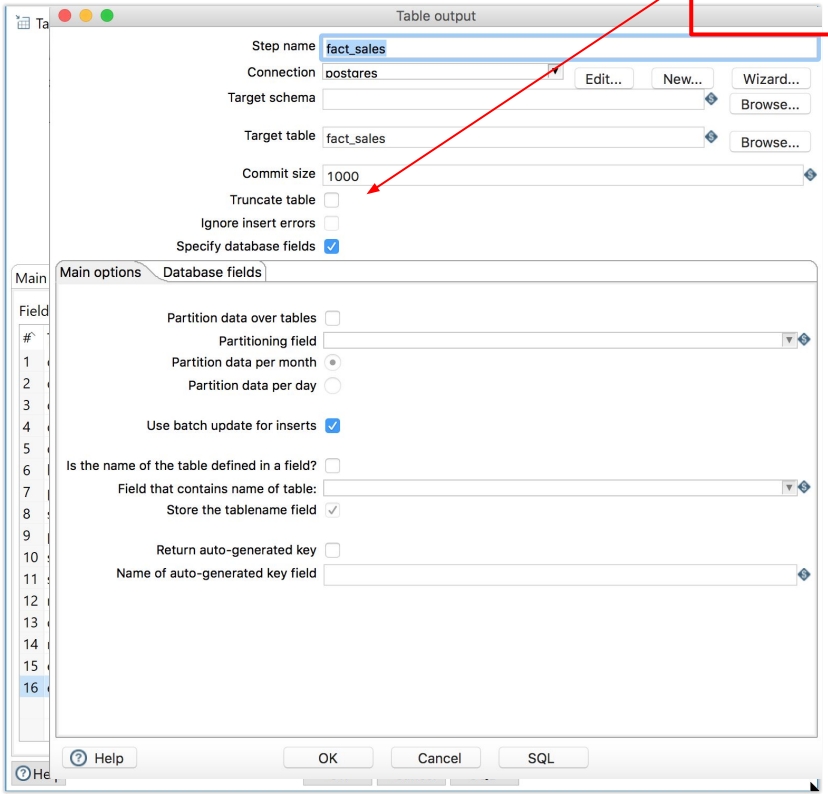
Values to return from the lookup table :

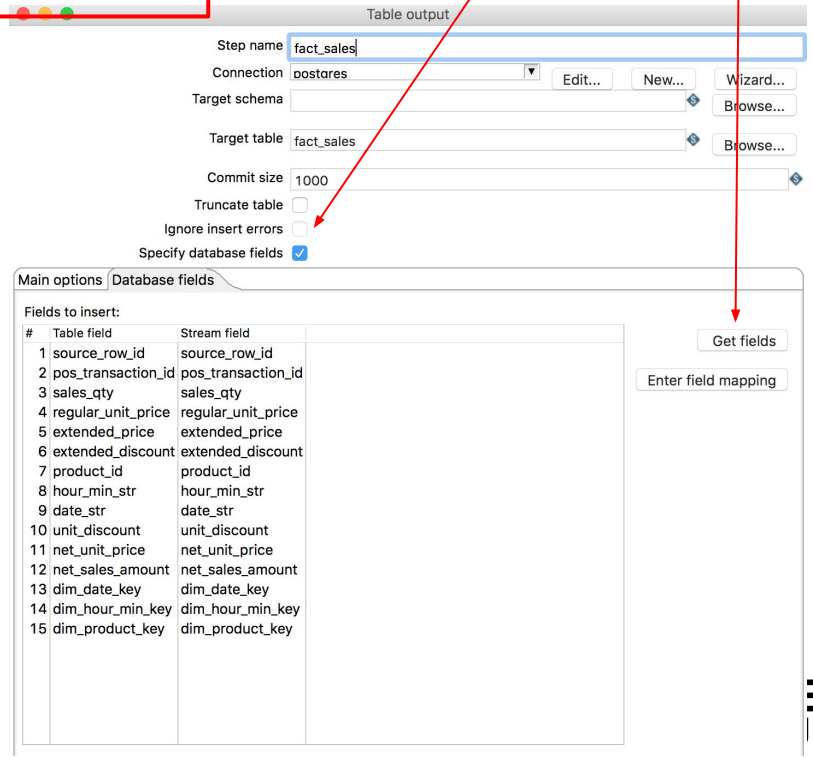
#	Field	New name	Default	Type
1	dim_product_key		0	Integer

Table output

WARNING:
Do Not truncate table

Specify database fields
Get fields





#	Table field	Stream field
1	source_row_id	source_row_id
2	pos_transaction_id	pos_transaction_id
3	sales_qty	sales_qty
4	regular_unit_price	regular_unit_price
5	extended_price	extended_price
6	extended_discount	extended_discount
7	product_id	product_id
8	hour_min_str	hour_min_str
9	date_str	date_str
10	unit_discount	unit_discount
11	net_unit_price	net_unit_price
12	net_sales_amount	net_sales_amount
13	dim_date_key	dim_date_key
14	dim_hour_min_key	dim_hour_min_key
15	dim_product_key	dim_product_key

Remove redundant fields

Main options | Database fields

Fields to insert:

#	Table field	Stream field
1	source_row_id	source_row_id
2	pos_transactio...	pos_transaction_id
3	sales_qty	sales_qty
4	regular_unit_price	regular_unit_price
5	extended_price	extended_price
6	extended_disc...	extended_discount
7	product_id	product_id
8	hour_min_str	hour_min_str
9	date_str	date_str
10	unit_discount	unit_discount
11	net_unit_price	net_unit_price
12	net_sales_amount	net_sales_amount
13	dim_date_key	dim_date_key
14	dim_hour_min_...	dim_hour_min_key
15	dim_product_key	dim_product_key

Get fields

ping

- Insert before this row
- Insert after this row
- Move up ⌘↑
- Move down ⌘↓
- Optimal Column size incl. header F3
- Optimal Column size excl. header F4
- Clear all
- Select all rows ⌘A
- Clear selection
- Filtered selection ⌘F
- Copy selected lines to clipboard ⌘C
- Paste clipboard to table ⌘V
- Cut selected lines ⌘X
- Delete selected lines
- Keep only selected lines ⌘K
- Copy field value to all rows

Run SQL to create fact_sales table

Verify/debug the transformation

Run the transformation

Check the metrics


stage_sales
date lookup
hour_min lookup
product lookup
fact_sales

Run Transformation #1

Execution Results

Execution History | Logging | Step Metrics | Performance Graph | Metrics | Preview data

#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
1	stage_sales	0	0	73168	73168	0	0	0	0	Finished	14.2s	5,166	-
2	date lookup	0	73168	73168	73168	0	0	0	0	Finished	16.2s	4,503	-
3	hour_min lookup	0	73168	73168	73168	0	0	0	0	Finished	16.3s	4,497	-
4	product lookup	0	73168	73168	73168	0	0	0	0	Finished	17.2s	4,242	-
5	fact_sales	0	73168	73168	0	73168	0	0	0	Finished	17.3s	4,229	-

ED
JBE

Run Transformation #2

?

Same rows written again into the fact table

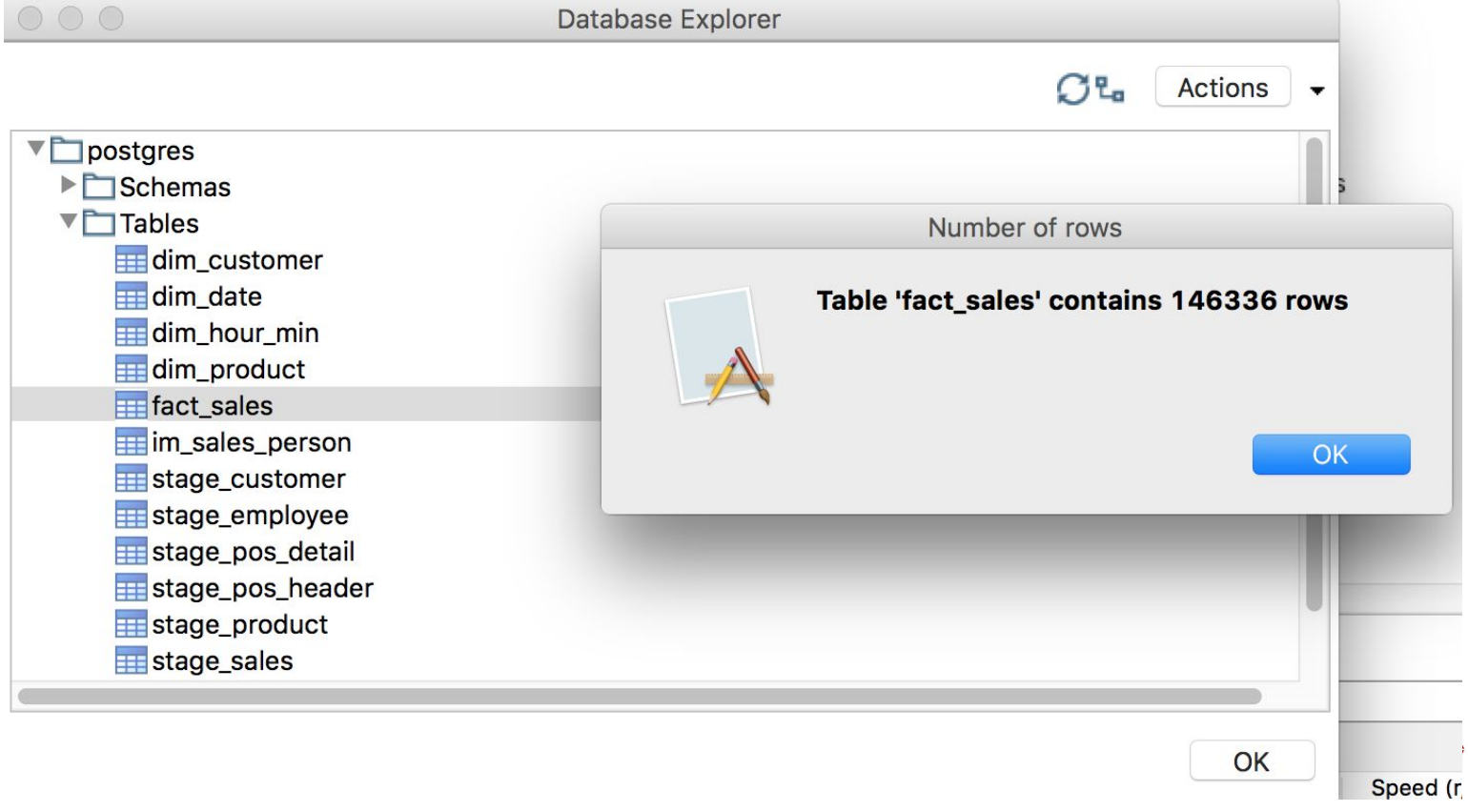
Execution Results

Execution History | Logging | Step Metrics | Performance Graph | Metrics | Preview data

#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
1	stage_sales	0	0	73168	73168	0	0	0	0	Finished	10.2s	7,193	-
2	date lookup	0	73168	73168	73168	0	0	0	0	Finished	11.6s	6,282	-
3	hour_min lookup	0	73168	73168	73168	0	0	0	0	Finished	11.8s	6,226	-
4	product lookup	0	73168	73168	73168	0	0	0	0	Finished	11.9s	6,172	-
5	fact_sales	0	73168	73168	0	73168	0	0	0	Finished	11.9s	6,152	-

RED CUBE

Fact table row count



RED CUBE

Do we need a fact table primary key?

- The requirement for a primary key in a fact table depends on the type (Transactional/Periodic snapshot/Accumulating Snapshot) of the fact table.

A row in a *transaction fact table* corresponds to a measurement event at a point in space and time. Atomic transaction grain fact tables are the most dimensional and expressive fact tables; this robust dimensionality enables the maximum slicing and dicing of transaction data. Transaction fact tables may be dense or sparse because rows exist only if measurements take place. These fact tables always contain a foreign key for each associated dimension, and optionally contain precise time stamps and degenerate dimension keys. The measured numeric facts must be consistent with the transaction grain.

-Kimbal

- Transactional facts which are never updated do not need primary keys.
- It does not make sense to amend a transaction at a retail shop which happened yesterday.
- **To summarize: We do not need one.**
  - **How to prevent duplicate entries?**
    - **Solution: Make sure same file is not processed twice**

If you really need one?

1. Surrogate key
2. Make a natural key of the fact table primary key. E.g. source_row_id
3. Depends entirely on the needs of the business users.

How to Create primary key in fact table?

The Not-so-elegant solution: Do it manually from Pgadmin/write sql

End of Task 2.2