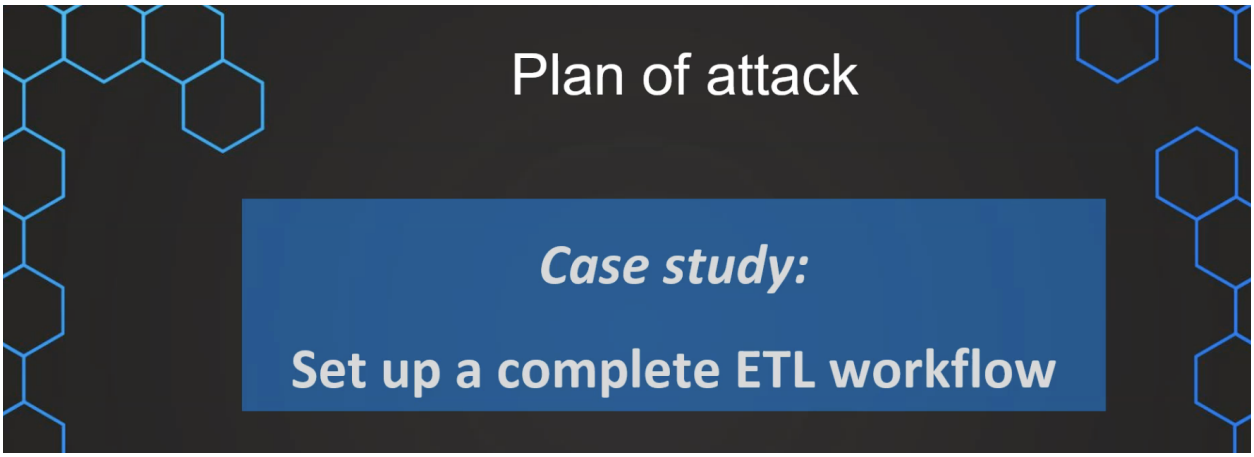
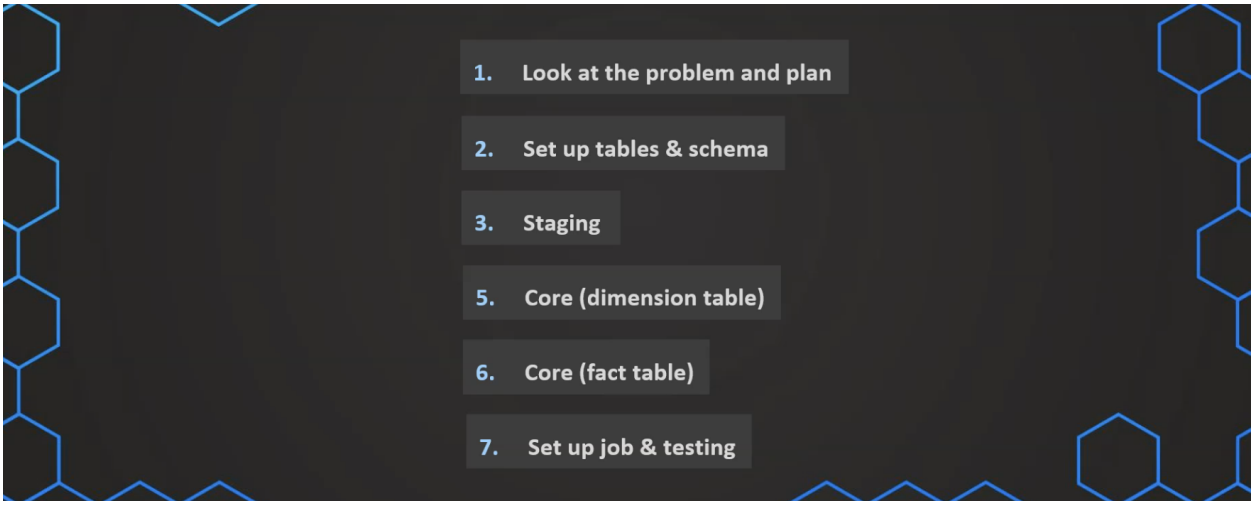
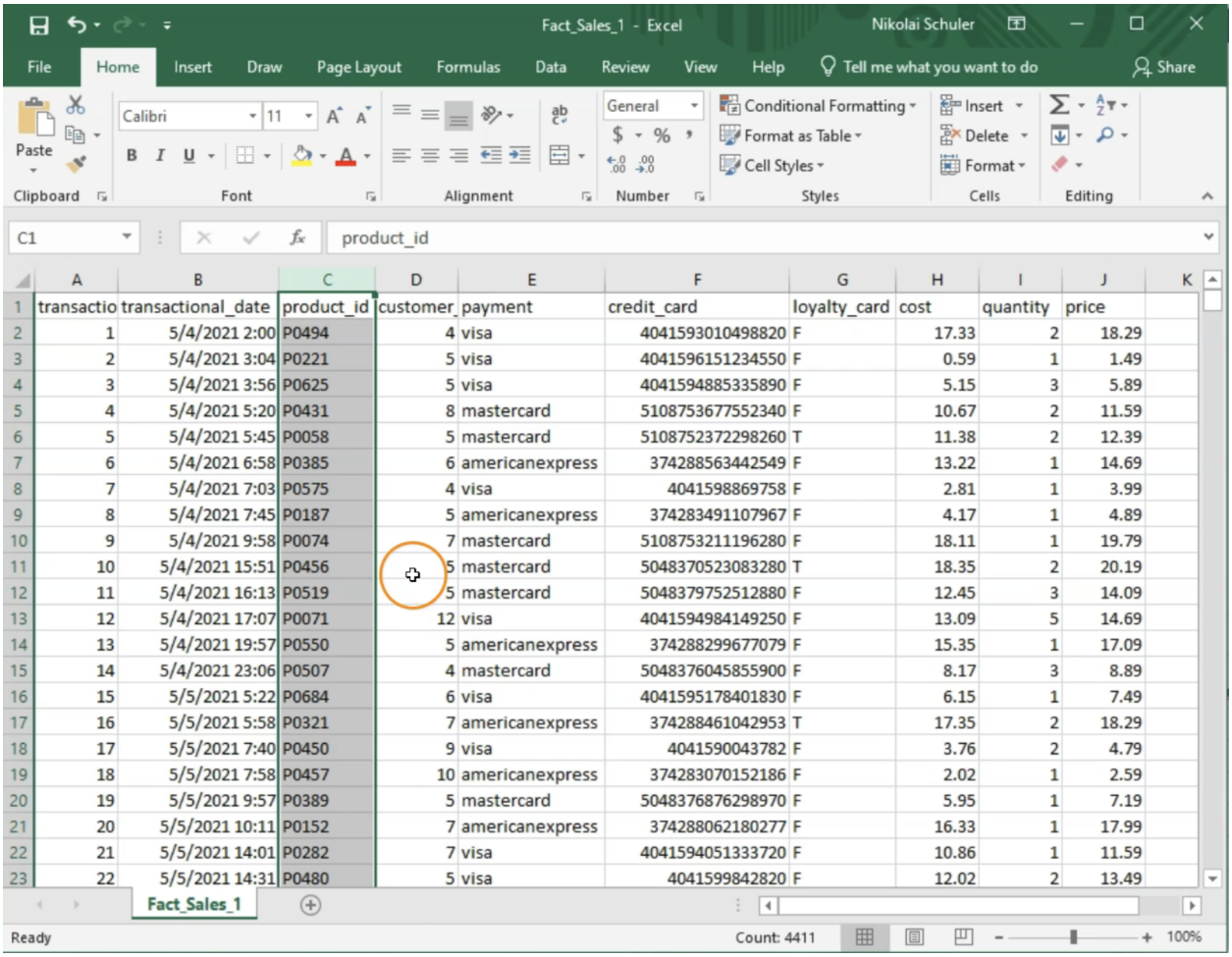
# 11. Creating a Datawarehouse-Project



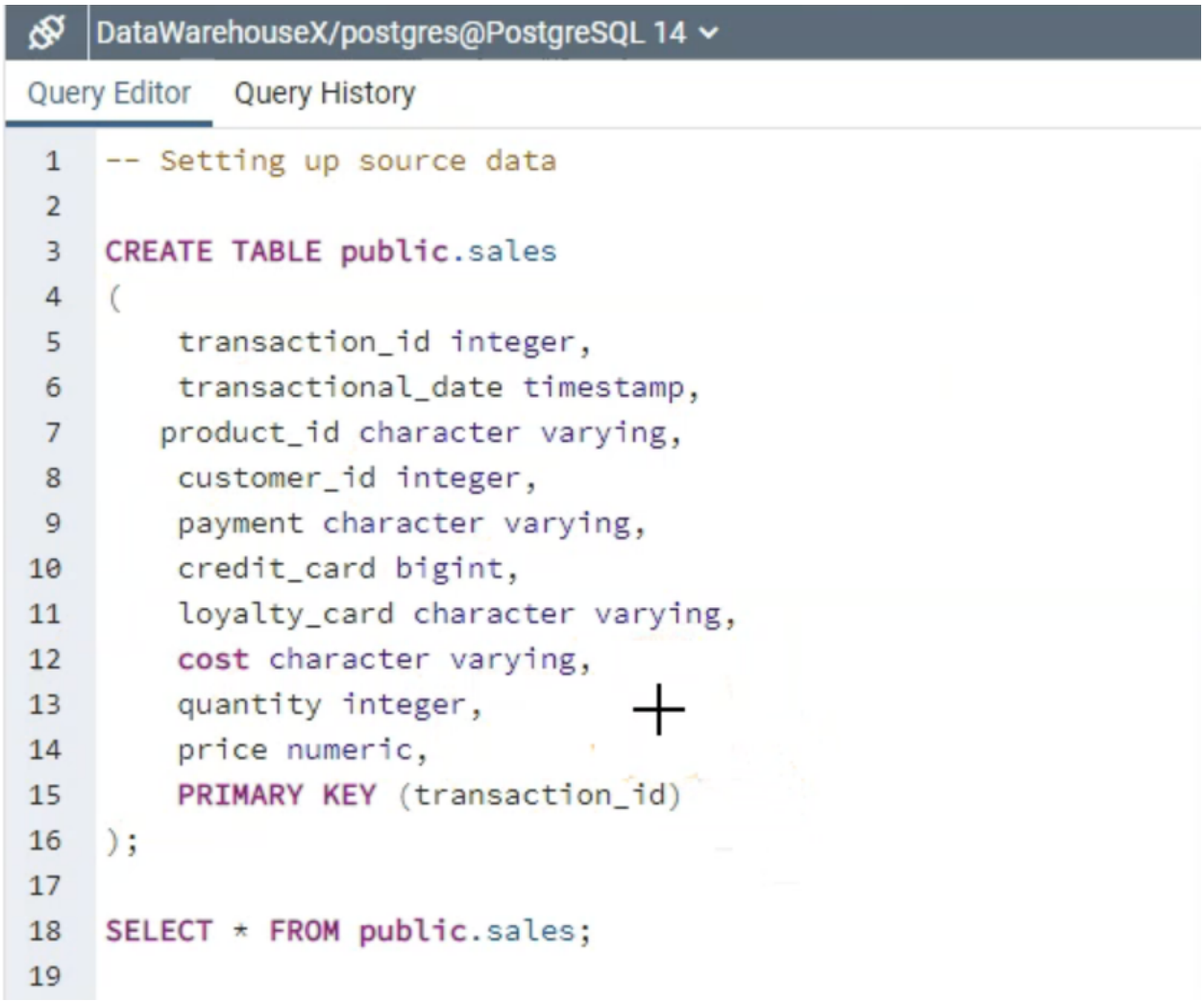

## Source data and table design

- Below is our sales data

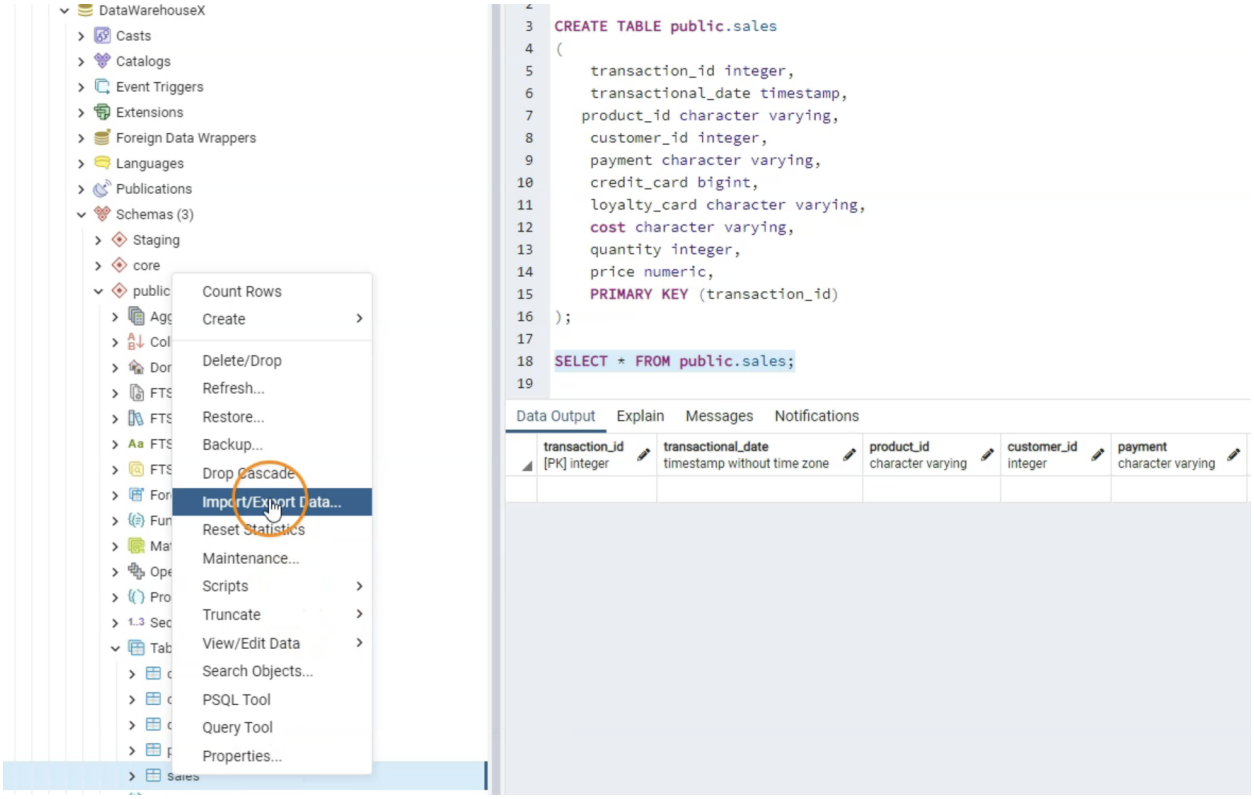


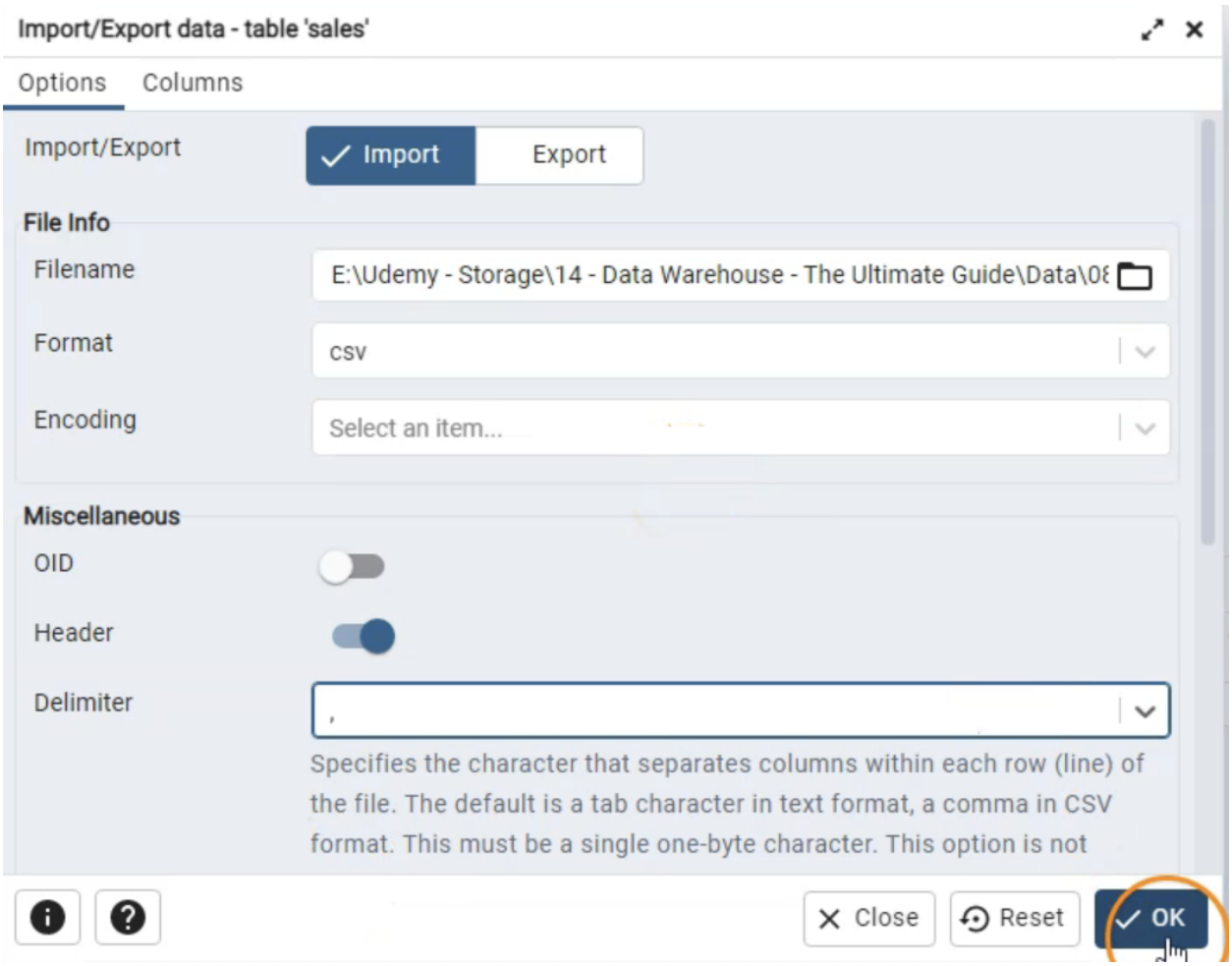

| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | transactio | transactional_date | product_id | customer_ | payment | credit_card | loyalty_card | cost | quantity | price | |
| 2 | 1 | 5/4/2021 2:00 | P0494 | 4 | visa | 4041593010498820 | F | 17.33 | 2 | 18.29 | |
| 3 | 2 | 5/4/2021 3:04 | P0221 | 5 | visa | 4041596151234550 | F | 0.59 | 1 | 1.49 | |
| 4 | 3 | 5/4/2021 3:56 | P0625 | 5 | visa | 4041594885335890 | F | 5.15 | 3 | 5.89 | |
| 5 | 4 | 5/4/2021 5:20 | P0431 | 8 | mastercard | 5108753677552340 | F | 10.67 | 2 | 11.59 | |
| 6 | 5 | 5/4/2021 5:45 | P0058 | 5 | mastercard | 5108752372298260 | T | 11.38 | 2 | 12.39 | |
| 7 | 6 | 5/4/2021 6:58 | P0385 | 6 | americanexpress | 374288563442549 | F | 13.22 | 1 | 14.69 | |
| 8 | 7 | 5/4/2021 7:03 | P0575 | 4 | visa | 4041598869758 | F | 2.81 | 1 | 3.99 | |
| 9 | 8 | 5/4/2021 7:45 | P0187 | 5 | americanexpress | 374283491107967 | F | 4.17 | 1 | 4.89 | |
| 10 | 9 | 5/4/2021 9:58 | P0074 | 7 | mastercard | 5108753211196280 | F | 18.11 | 1 | 19.79 | |
| 11 | 10 | 5/4/2021 15:51 | P0456 | 5 | mastercard | 5048370523083280 | T | 18.35 | 2 | 20.19 | |
| 12 | 11 | 5/4/2021 16:13 | P0519 | 5 | mastercard | 5048379752512880 | F | 12.45 | 3 | 14.09 | |
| 13 | 12 | 5/4/2021 17:07 | P0071 | 12 | visa | 4041594984149250 | F | 13.09 | 5 | 14.69 | |
| 14 | 13 | 5/4/2021 19:57 | P0550 | 5 | americanexpress | 374288299677079 | F | 15.35 | 1 | 17.09 | |
| 15 | 14 | 5/4/2021 23:06 | P0507 | 4 | mastercard | 5048376045855900 | F | 8.17 | 3 | 8.89 | |
| 16 | 15 | 5/5/2021 5:22 | P0684 | 6 | visa | 4041595178401830 | F | 6.15 | 1 | 7.49 | |
| 17 | 16 | 5/5/2021 5:58 | P0321 | 7 | americanexpress | 374288461042953 | T | 17.35 | 2 | 18.29 | |
| 18 | 17 | 5/5/2021 7:40 | P0450 | 9 | visa | 4041590043782 | F | 3.76 | 2 | 4.79 | |
| 19 | 18 | 5/5/2021 7:58 | P0457 | 10 | americanexpress | 374283070152186 | F | 2.02 | 1 | 2.59 | |
| 20 | 19 | 5/5/2021 9:57 | P0389 | 5 | mastercard | 5048376876298970 | F | 5.95 | 1 | 7.19 | |
| 21 | 20 | 5/5/2021 10:11 | P0152 | 7 | americanexpress | 374288062180277 | F | 16.33 | 1 | 17.99 | |
| 22 | 21 | 5/5/2021 14:01 | P0282 | 7 | visa | 4041594051333720 | F | 10.86 | 1 | 11.59 | |
| 23 | 22 | 5/5/2021 14:31 | P0480 | 5 | visa | 4041599842820 | F | 12.02 | 2 | 13.49 | |

- Let us load this data in public schema which will act as source data




- Now we will have empty table, we should now import the data from CSV file


DataWarehouseX
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas (3)
Staging
core
public
Count Rows
Create
Delete/Drop
Refresh...
Restore...
Backup...
Drop Cascade
Import/Export Data...
Reset Statistics
Maintenance...
Scripts
Truncate
View/Edit Data
Search Objects...
PSQL Tool
Query Tool
Properties...
sales
CREATE TABLE public.sales
(
transaction_id integer,
transactional_date timestamp,
product_id character varying,
customer_id integer,
payment character varying,
credit_card bigint,
loyalty_card character varying,
cost character varying,
quantity integer,
price numeric,
PRIMARY KEY (transaction_id)
);
SELECT * FROM public.sales;
Data Output
Explain
Messages
Notifications
transaction_id
[PK] integer
transactional_date
timestamp without time zone
product_id
character varying
customer_id
integer
payment
character varying



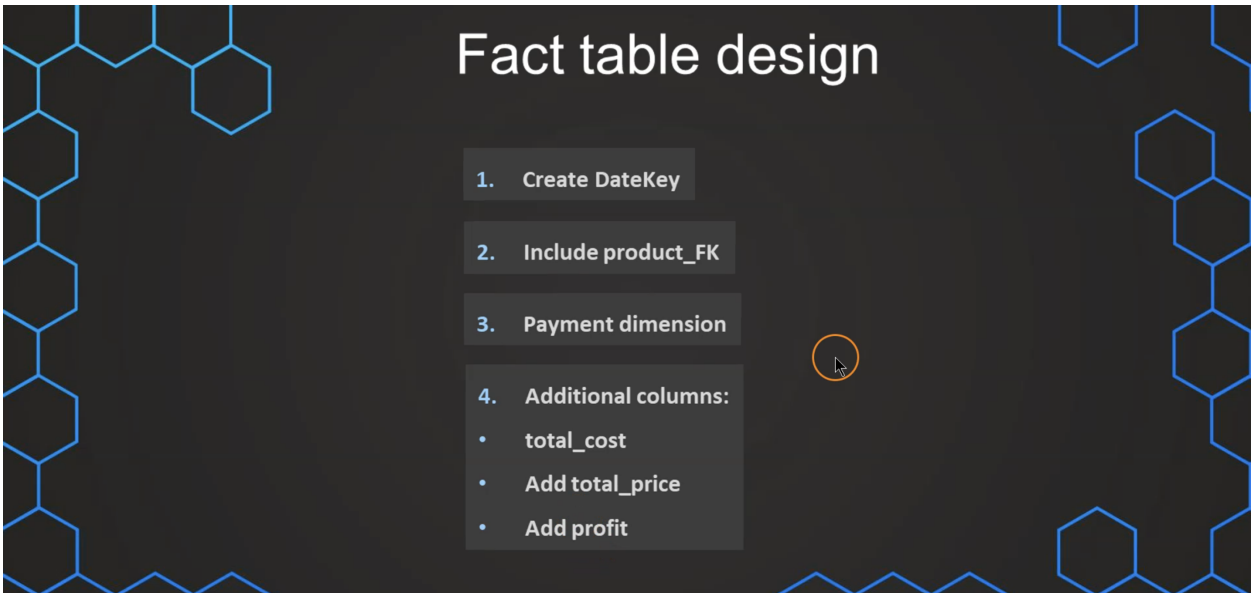

```
18  SELECT * FROM public.sales;
19
```

Data Output | Explain | Messages | Notifications

| | transaction_id [PK] integer | transactional_date timestamp without time zone | product_id character varying | customer_id integer | payment character varying | credit_card bigint | loyalty_card character varying | cost character varying | quantity integer | price numeric |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2021-05-04 02:00:00 | P0494 | 4 | visa | 41593010498829 | F | 17.33 | 2 | 18.29 |
| 2 | 2 | 2021-05-04 03:04:00 | P0221 | 5 | visa | 41596151234556 | F | 0.59 | 1 | 1.49 |
| 3 | 3 | 2021-05-04 03:56:00 | P0625 | 5 | visa | 41594885335898 | F | 5.15 | 3 | 5.89 |
| 4 | 4 | 2021-05-04 05:20:00 | P0431 | 8 | mastercard | 5375677552345 | F | 10.67 | 2 | 11.59 |
| 5 | 5 | 2021-05-04 05:45:00 | P0058 | 5 | mastercard | 5372298261 | T | 11.38 | 2 | 12.39 |
| 6 | 6 | 2021-05-04 06:58:00 | P0385 | 6 | americanexpress | 74288563442549 | F | 13.22 | 1 | 14.69 |
| 7 | 7 | 2021-05-04 07:03:00 | P0575 | 4 | visa | 4041598869758 | F | 2.81 | 1 | 3.99 |
| 8 | 8 | 2021-05-04 07:45:00 | P0187 | 5 | americanexpress | 74283491107967 | F | 4.17 | 1 | 4.89 |
| 9 | 9 | 2021-05-04 09:58:00 | P0074 | 7 | mastercard | 5375211196286 | F | 18.11 | 1 | 19.79 |
| 10 | 10 | 2021-05-04 15:51:00 | P0456 | 5 | mastercard | 48370523083285 | T | 18.35 | 2 | 20.19 |
| 11 | 11 | 2021-05-04 16:13:00 | P0519 | 5 | mastercard | 48379752512880 | F | 12.45 | 3 | 14.09 |
| 12 | 12 | 2021-05-04 17:07:00 | P0071 | 12 | visa | 41594984149257 | F | 13.09 | 5 | 14.69 |
| 13 | 13 | 2021-05-04 19:57:00 | P0550 | 5 | americanexpress | 74288299677079 | F | 15.35 | 1 | 17.09 |
| 14 | 14 | 2021-05-04 23:06:00 | P0507 | 4 | mastercard | 48376045855902 | F | 8.17 | 3 | 8.89 |
| 15 | 15 | 2021-05-05 05:22:00 | P0684 | 6 | visa | 41595178401834 | F | 6.15 | 1 | 7.49 |

- Here transaction_id which is natural key from the source is already a good key which is integer and contigous which is equivalent to surrogate key so we need not create a new surrogate key

- We also have transactional_date which can be used for delta column
- Product_id is natural key in this fact table, we need to bring the surrogate key
- customer_id is already good if we are going to set up a customer dimension
- payment and loyality_card looks like junk dimension and we have only 6-8 combination
- credit_card we will leave it as it is in fact table as it is degenerate dimension
- cost, quantity, price for each product are the measures




| transaction_id integer | transactional_date timestamp without time zone | transactional_date_fk numeric | product_id character varying | product_fk integer | payment_fk integer | customer_id integer | credit_card bigint | cost numeric | quantity integer | price numeric | total_price numeric | total_cost numeric | profit numeric |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2021-05-04 02:00:00 | 20210504 | P0494 | 13519 | 1 | 4 | 4041593010498829 | 17.33 | 2 | 18.29 | 36.58 | 34.66 | 1.92 |
| 2 | 2021-05-04 03:04:00 | 20210504 | P0221 | 13267 | 1 | 5 | 4041596151234556 | 0.59 | 1 | 1.49 | 1.49 | 0.59 | 0.90 |
| 3 | 2021-05-04 03:56:00 | 20210504 | P0625 | 13643 | 1 | 5 | 4041594885335898 | 5.15 | 3 | 5.89 | 17.67 | 15.45 | 2.22 |
| 4 | 2021-05-04 05:20:00 | 20210504 | P0431 | 13461 | 6 | 8 | 5108753677552345 | 10.67 | 2 | 11.59 | 23.18 | 21.34 | 1.84 |
| 5 | 2021-05-04 05:45:00 | 20210504 | P0058 | 13113 | 4 | 5 | 5108752372298261 | 11.38 | 2 | 12.39 | 24.78 | 22.76 | 2.02 |
| 6 | 2021-05-04 06:58:00 | 20210504 | P0385 | 13416 | 8 | 6 | 374288563442549 | 13.22 | 1 | 14.69 | 14.69 | 13.22 | 1.47 |
| 7 | 2021-05-04 07:03:00 | 20210504 | P0575 | 13596 | 1 | 4 | 4041598869758 | 2.81 | 1 | 3.99 | 3.99 | 2.81 | 1.18 |

# Setting up tables in database

- In the staging layer we will just have a copy of data and schema from source
- In core layer we will create the fact and dimensions and any additional columns

```
-- Setting up sales fact table for staging

CREATE TABLE "Staging".sales
(
    transaction_id integer,
    transactional_date timestamp,
    product_id character varying,
    customer_id integer,
    payment character varying,
    credit_card bigint,
    loyalty_card character varying,
    cost numeric,
    quantity integer,
    price numeric,
    PRIMARY KEY (transaction_id)
);
```

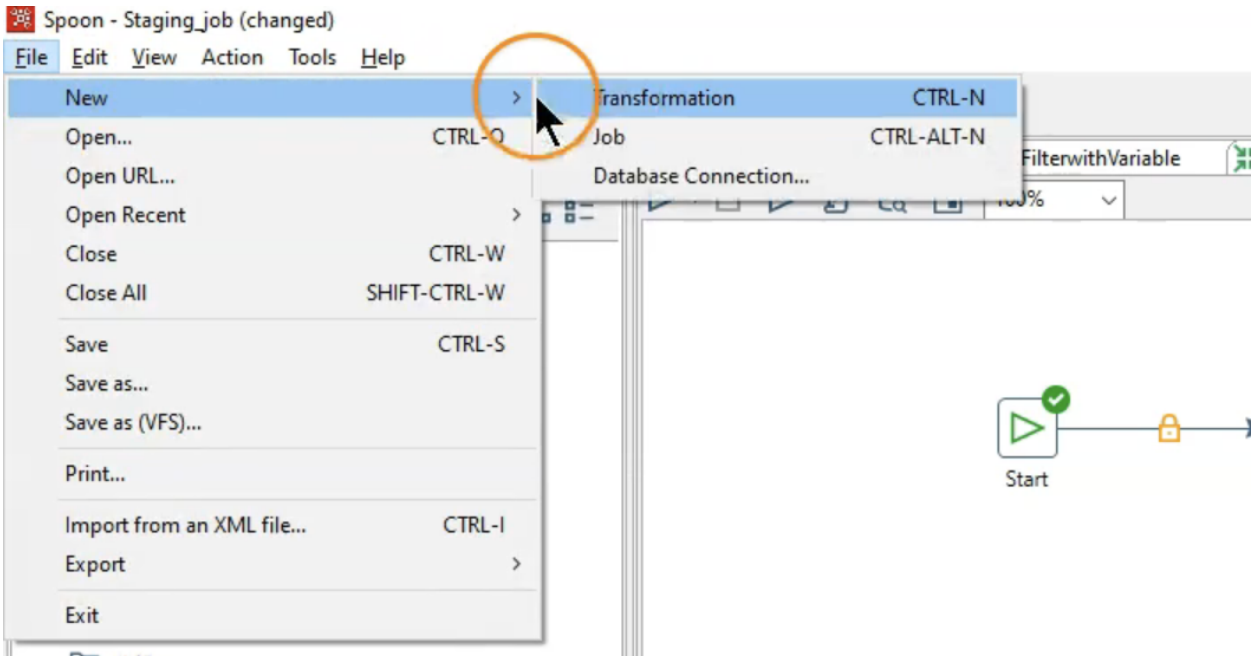
```
-- Setting up sales fact table for core

CREATE TABLE core.sales
(
    transaction_id integer,
    transactional_date timestamp,
    transactional_date_fk bigint,
    product_id character varying,
    product_fk integer,
    customer_id integer,
    payment_fk integer,
    credit_card bigint,
    cost numeric,
    quantity integer,
    price numeric,
    total_cost numeric,
    total_price numeric,
```

```
    profit numeric,
    PRIMARY KEY (transaction_id)
);
```

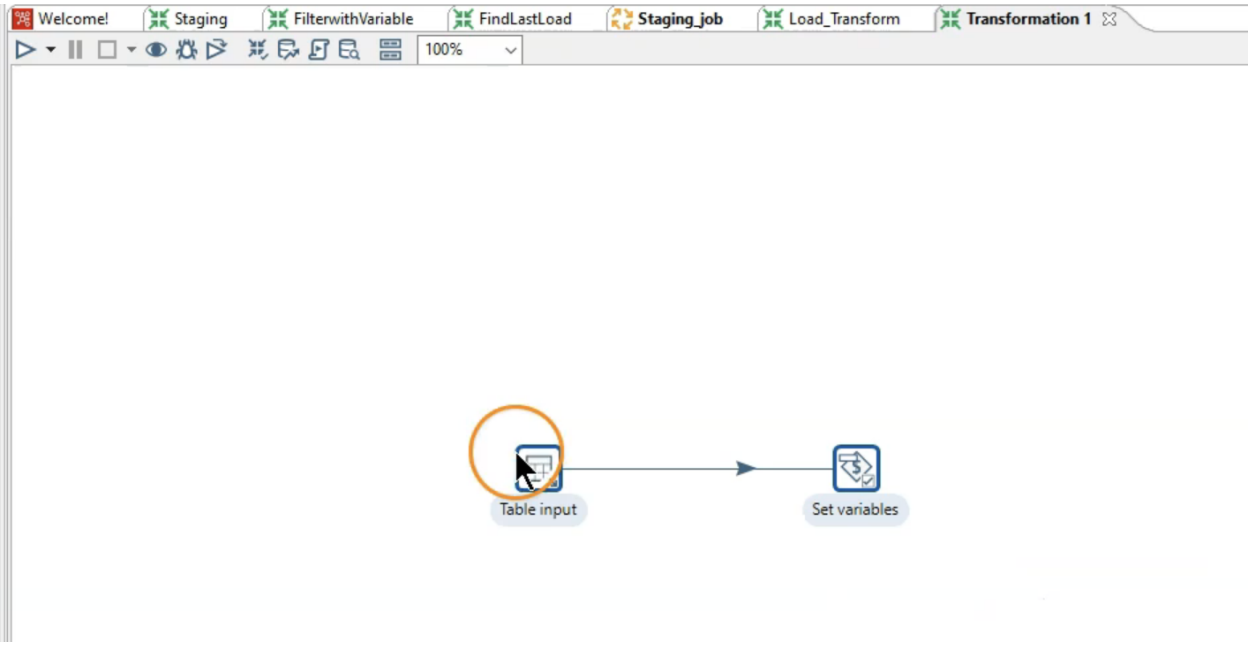
```
-- Setting up sales payment tables for staging and core

CREATE TABLE core.dim_payment
(
    payment_pk integer NOT NULL GENERATED BY DEFAULT AS IDENT
ITY ( INCREMENT 1 START 1 ),
    payment character varying,
    loyalty_card character varying,
    PRIMARY KEY (payment_pk)
);
```

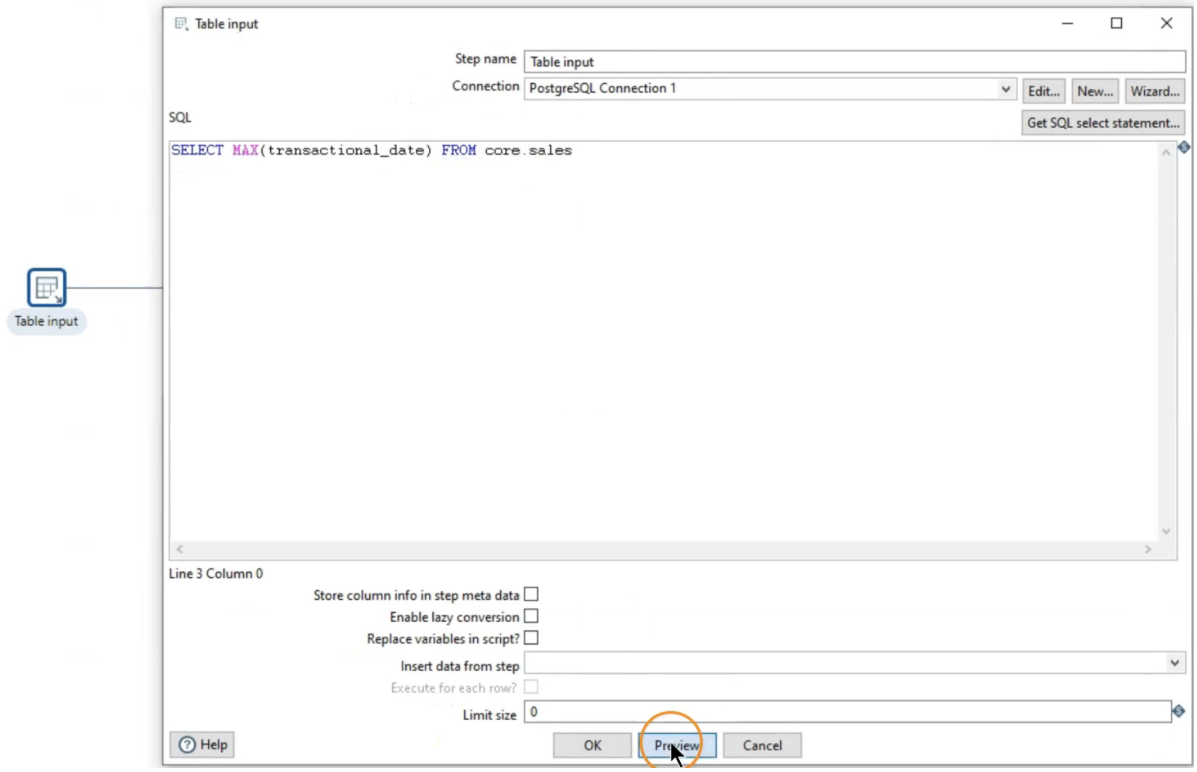
# Staging: Sales Fact

- Set up new transformation in the ETL tool. Go to File>New>Transformation



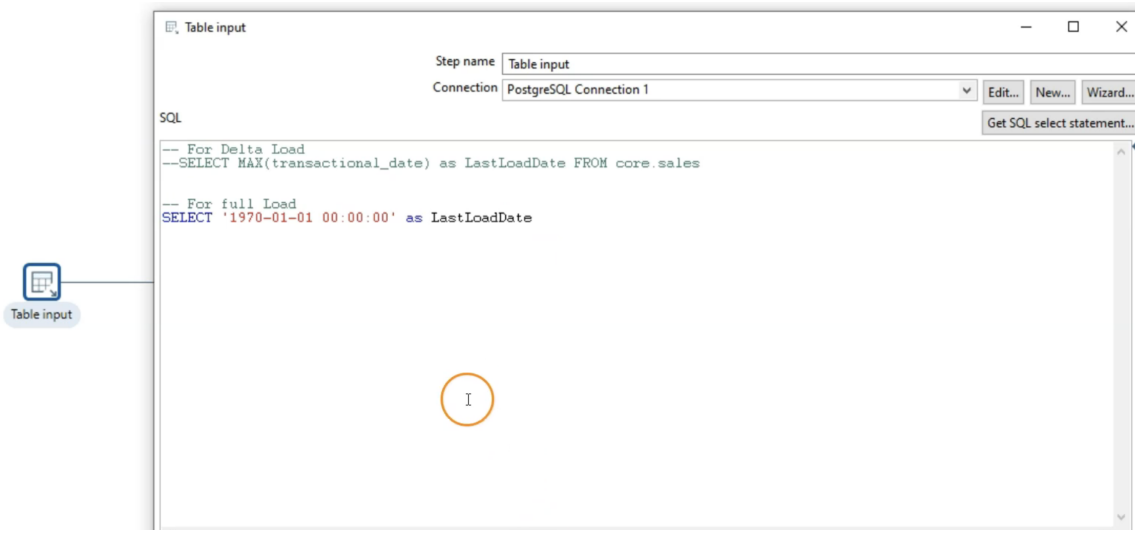
- We will first set up delta logic which is create a variable and get the highest transactional_date from the core layer



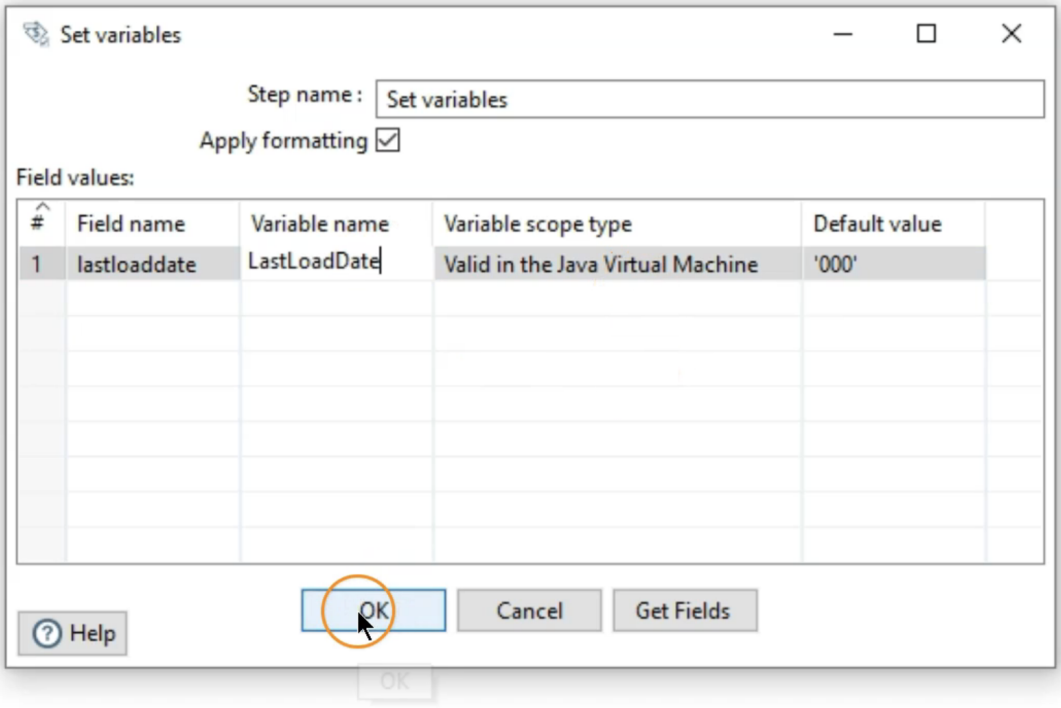



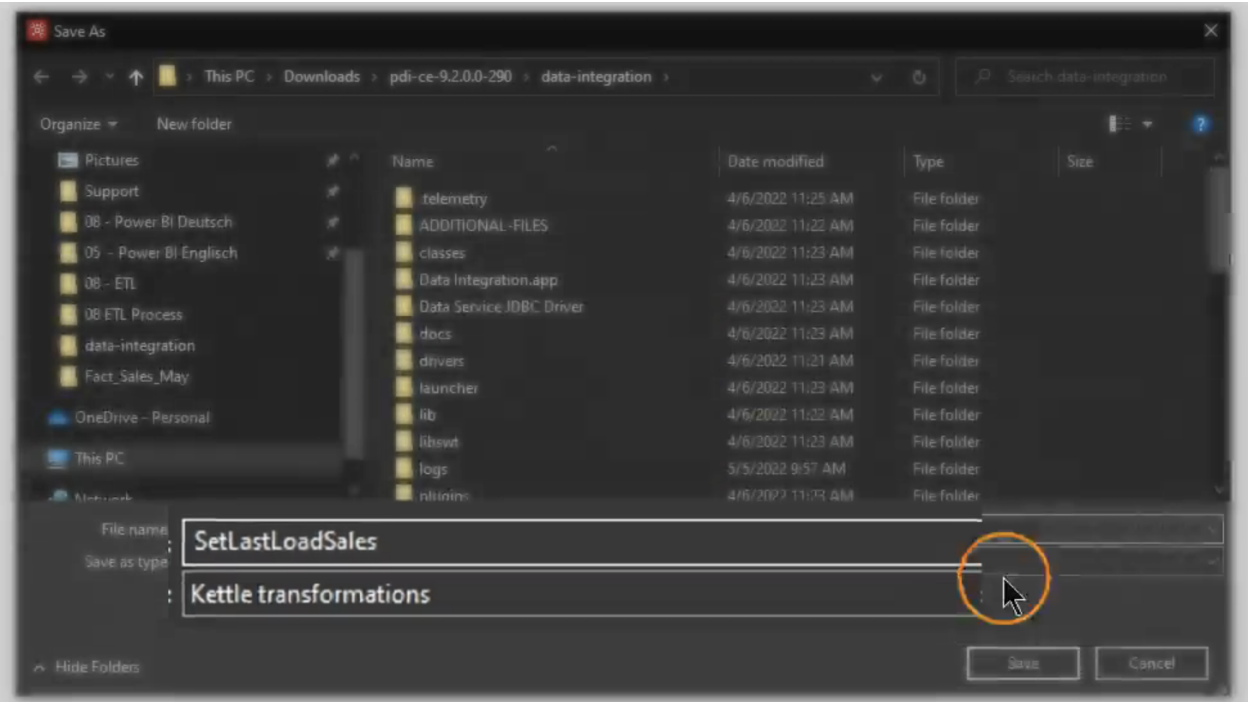

- Since there is no data yet in core layer we will get null


- Let us rename the variable to something meaningful like  LastLoadDate
- For Initial load we do not want null instead of can set up a dummy variable and later comment it for delta load
- We will uncomment the incremental load SQL code later

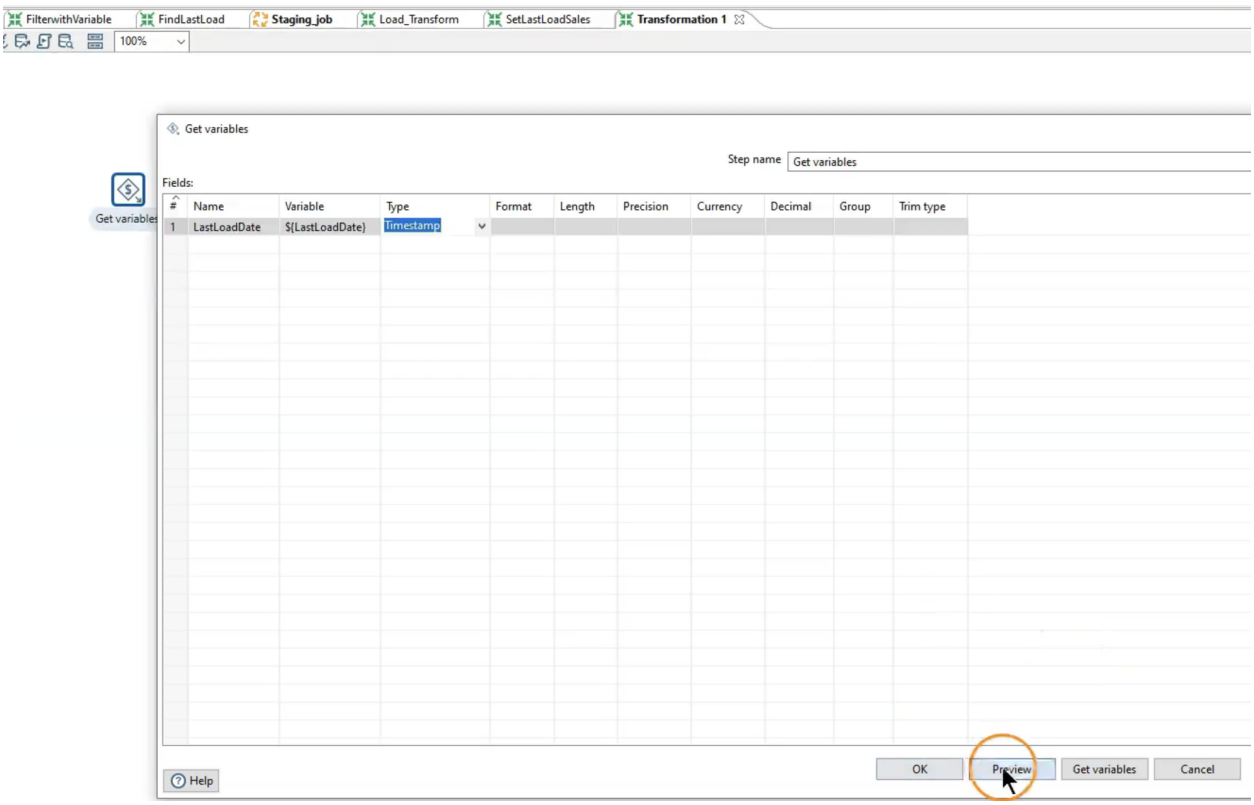


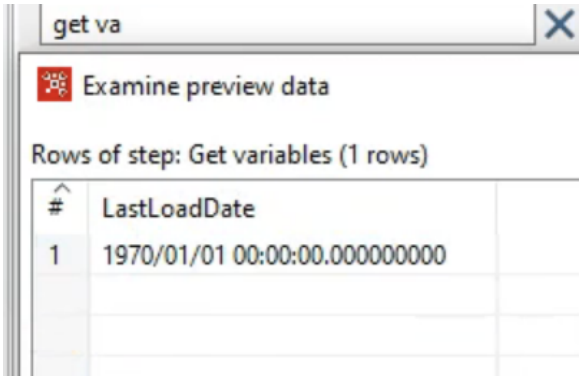

Examine preview data
Rows of step: Table input (1 rows)
# lastloaddate
1 1970-01-01 00:00:00


Set variables
Step name : Set variables
Apply formatting
Field values:
# Field name Variable name Variable scope type Default value
1 lastloaddate LastLoadDate Valid in the Java Virtual Machine '000'
Help
OK
Cancel
Get Fields




- Let’s create new transformation to get the last load date value and load the data
- Create a new transformation for it. FIle> New> Transformation
- Get the value using ${LastLoadDate} and set the data type as timestamp


FilterwithVariable
FindLastLoad
Staging_job
Load_Transform
SetLastLoadSales
Transformation 1
100%
Get variables
Step name
Get variables
Fields:
# Name Variable Type Format Length Precision Currency Decimal Group Trim type
1 LastLoadDate ${LastLoadDate} Timestamp
Help
OK
Preview
Get variables
Cancel


get va
Examine preview data
Rows of step: Get variables (1 rows)
# LastLoadDate
1 1970/01/01 00:00:00.000000000

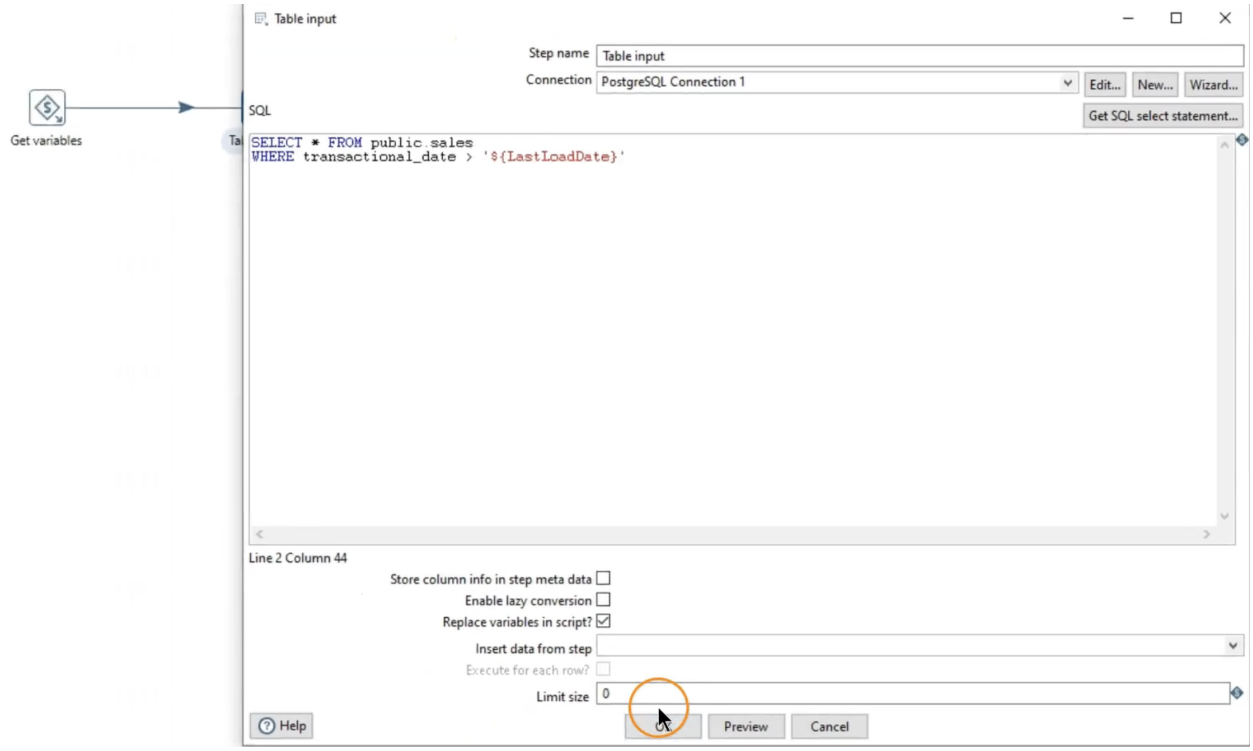
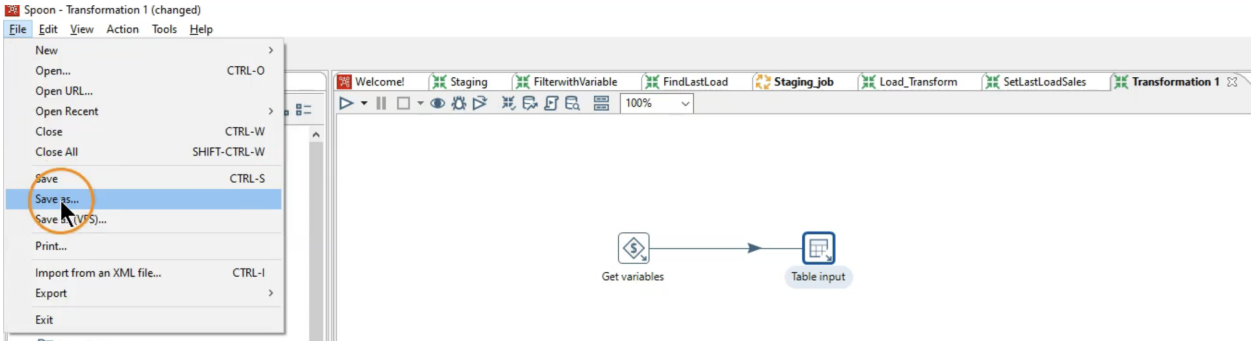
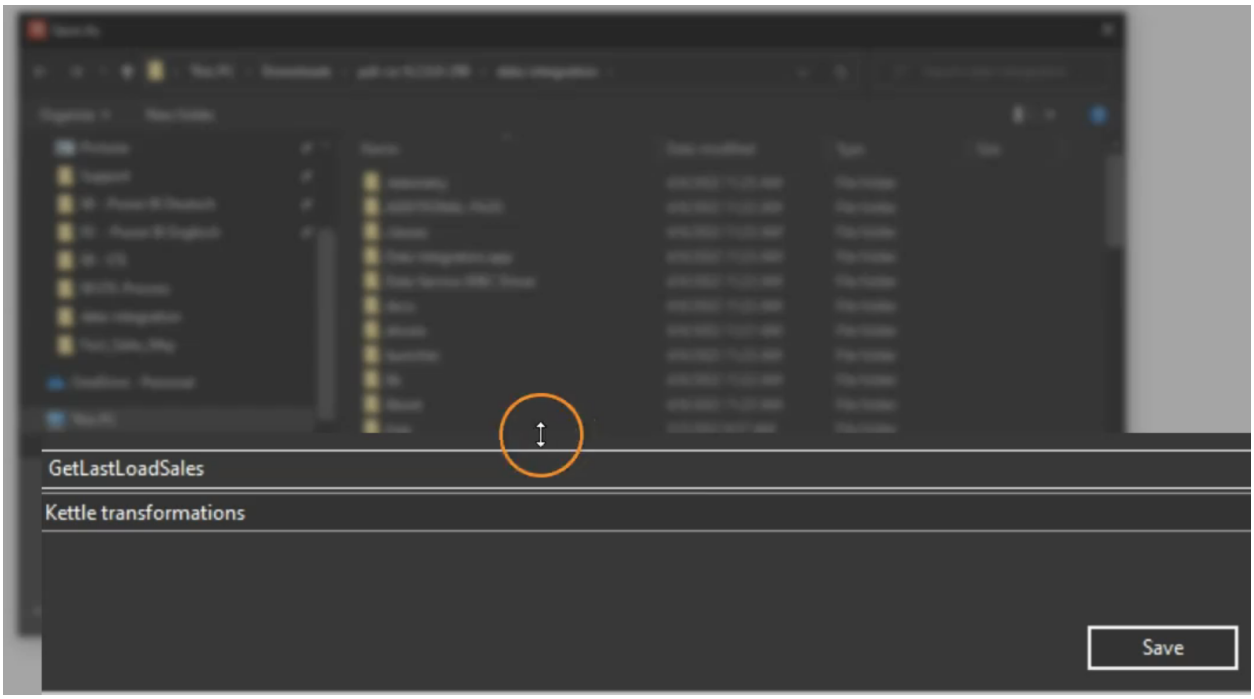


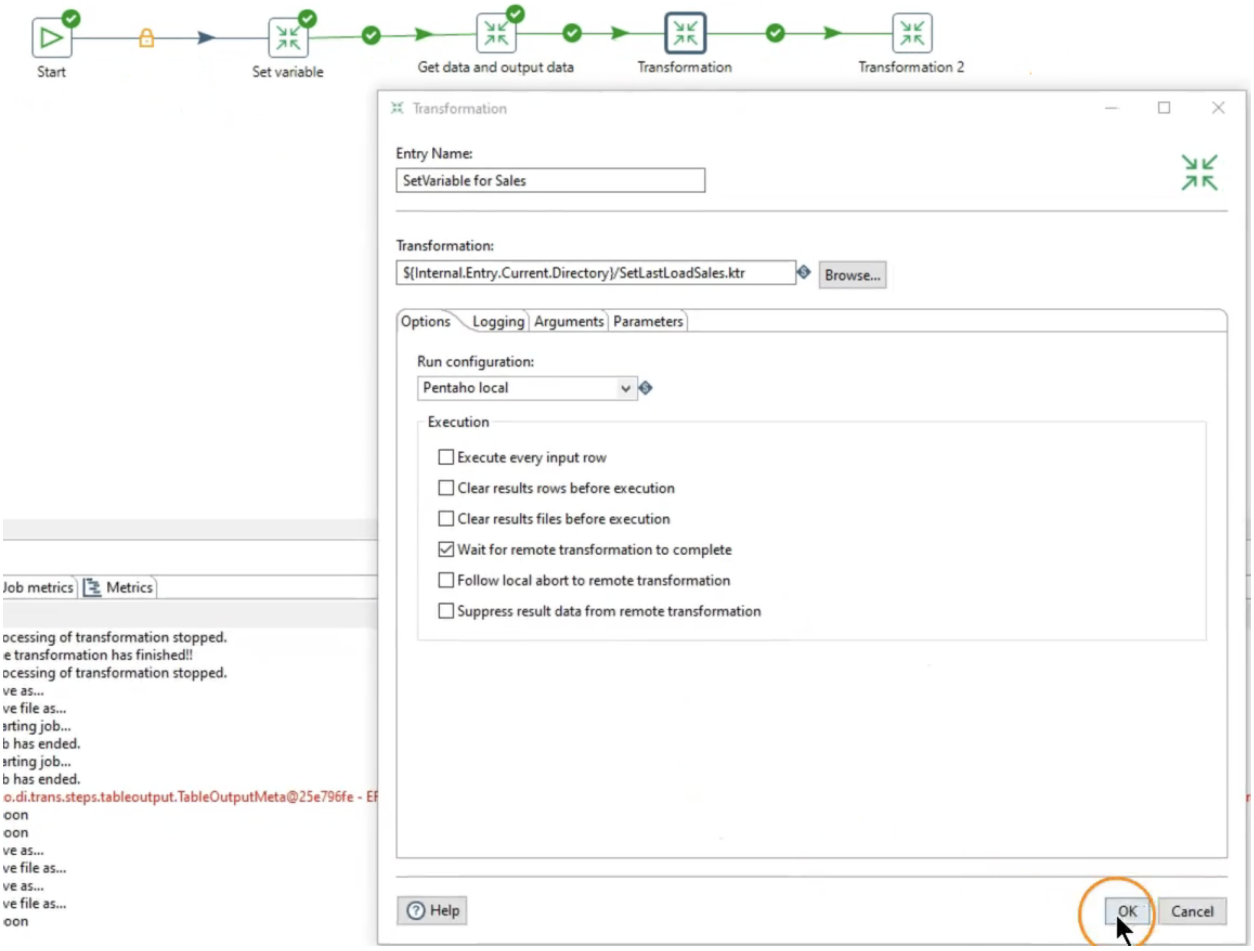
Table input
Step name
Table input
Connection
PostgreSQL Connection 1
Edit...
New...
Wizard...
SQL
Get SQL select statement...
Get variables
SELECT * FROM public.sales
WHERE transactional_date > '${LastLoadDate}'
Line 2 Column 44
Store column info in step meta data
Enable lazy conversion
Replace variables in script?
Insert data from step
Execute for each row?
Limit size
0
Help
Preview
Cancel

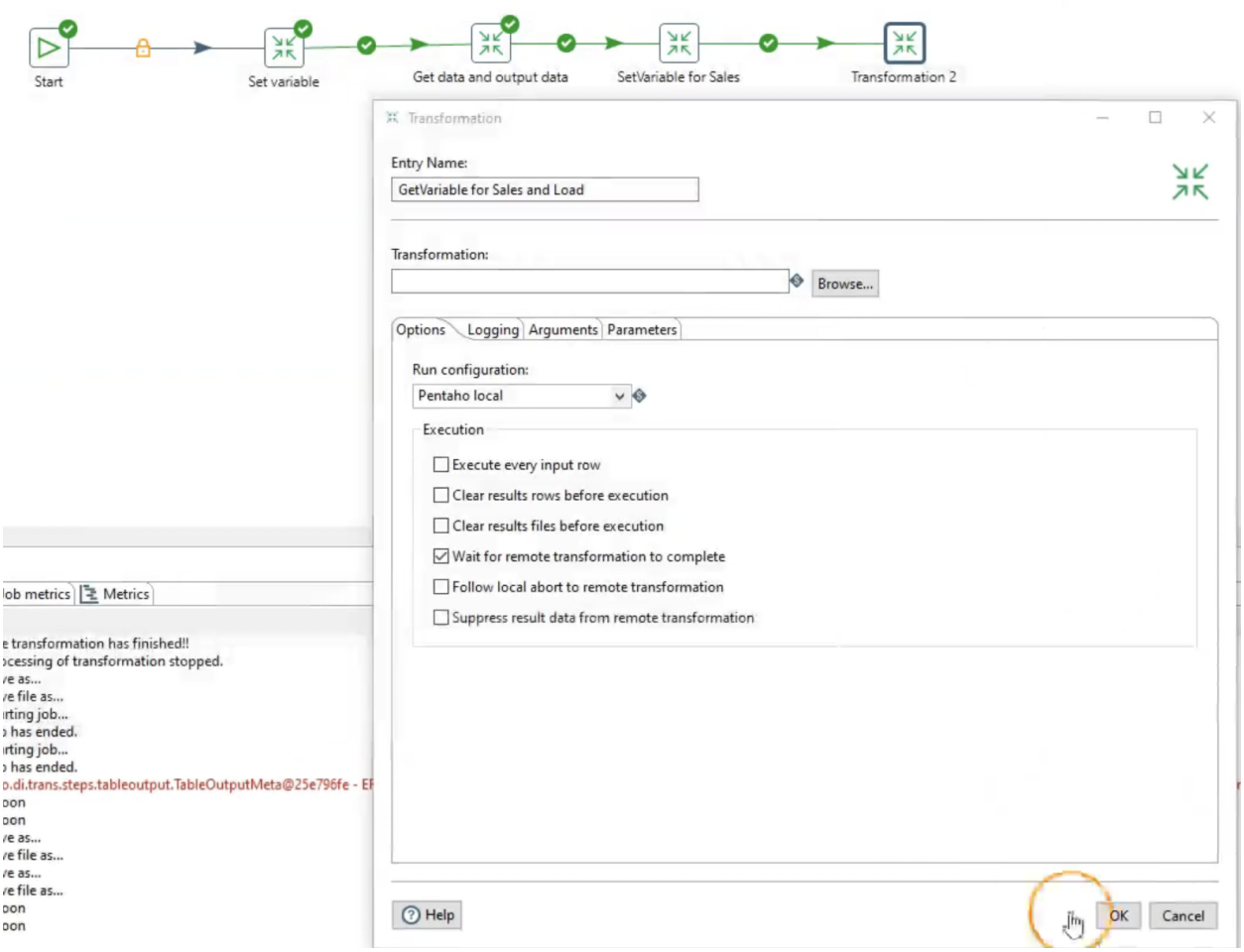

Spoon - Transformation 1 (changed)
File
Edit
View
Action
Tools
Help
New
Open... CTRL-O
Open URL...
Open Recent
Close CTRL-W
Close All SHIFT-CTRL-W
Save CTRL-S
Save as...
Print...
Import from an XML file... CTRL-I
Export
Exit
Welcome!
Staging
FilterwithVariable
FindLastLoad
Staging_job
Load_Transform
SetLastLoadSales
Transformation 1
100%
Get variables
Table input

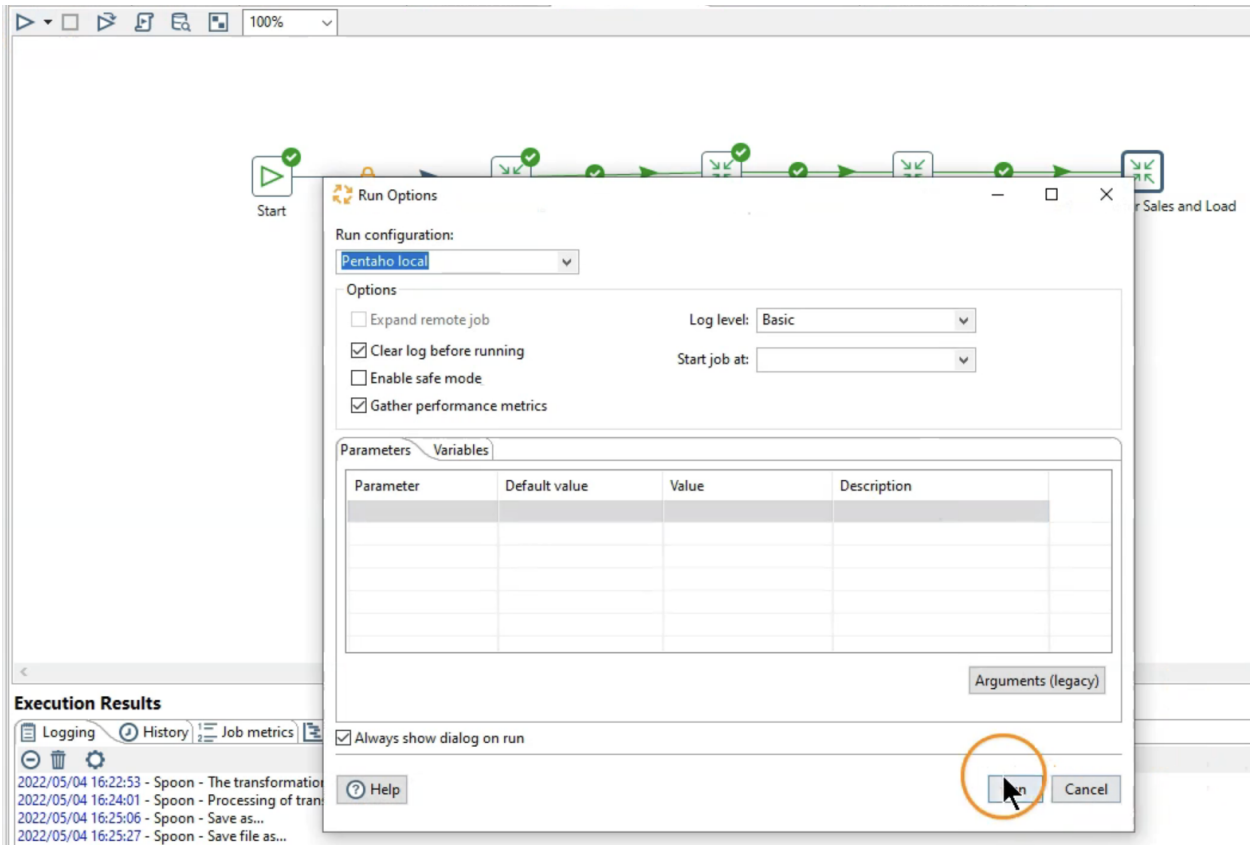


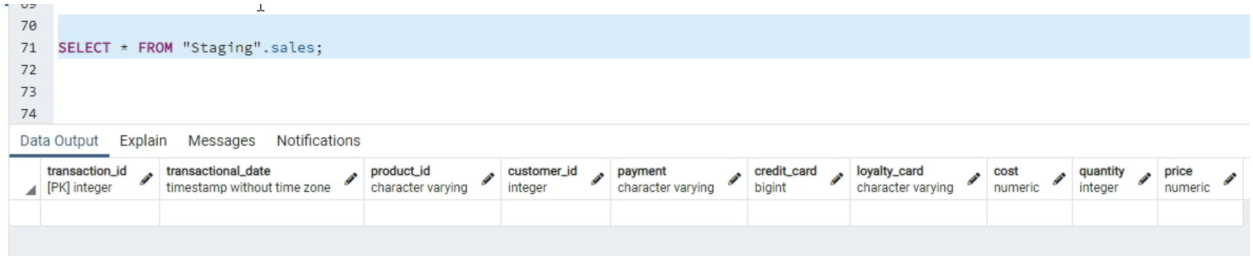

# Staging jobs and fixing problems

- With 2 transformations set up SetLastLoadSales and GetLastLoadSales
- Add 2 transformation in the Job and select respective saves transformation

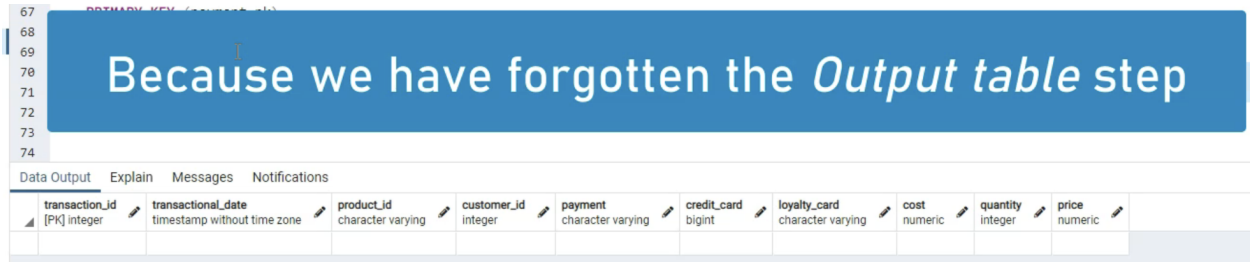

Start
Set variable
Get data and output data
Transformation
Transformation 2
Transformation
Entry Name:
SetVariable for Sales
Transformation:
${Internal.Entry.Current.Directory}/SetLastLoadSales.ktr
Browse...
Options
Logging
Arguments
Parameters
Run configuration:
Pentaho local
Execution
Execute every input row
Clear results rows before execution
Clear results files before execution
Wait for remote transformation to complete
Follow local abort to remote transformation
Suppress result data from remote transformation
Help
OK
Cancel
Job metrics
Metrics


Start
Set variable
Get data and output data
SetVariable for Sales
Transformation 2
Transformation
Entry Name:
GetVariable for Sales and Load
Transformation:
Browse...
Options
Logging
Arguments
Parameters
Run configuration:
Pentaho local
Execution
Execute every input row
Clear results rows before execution
Clear results files before execution
Wait for remote transformation to complete
Follow local abort to remote transformation
Suppress result data from remote transformation
Help
OK
Cancel
Metrics

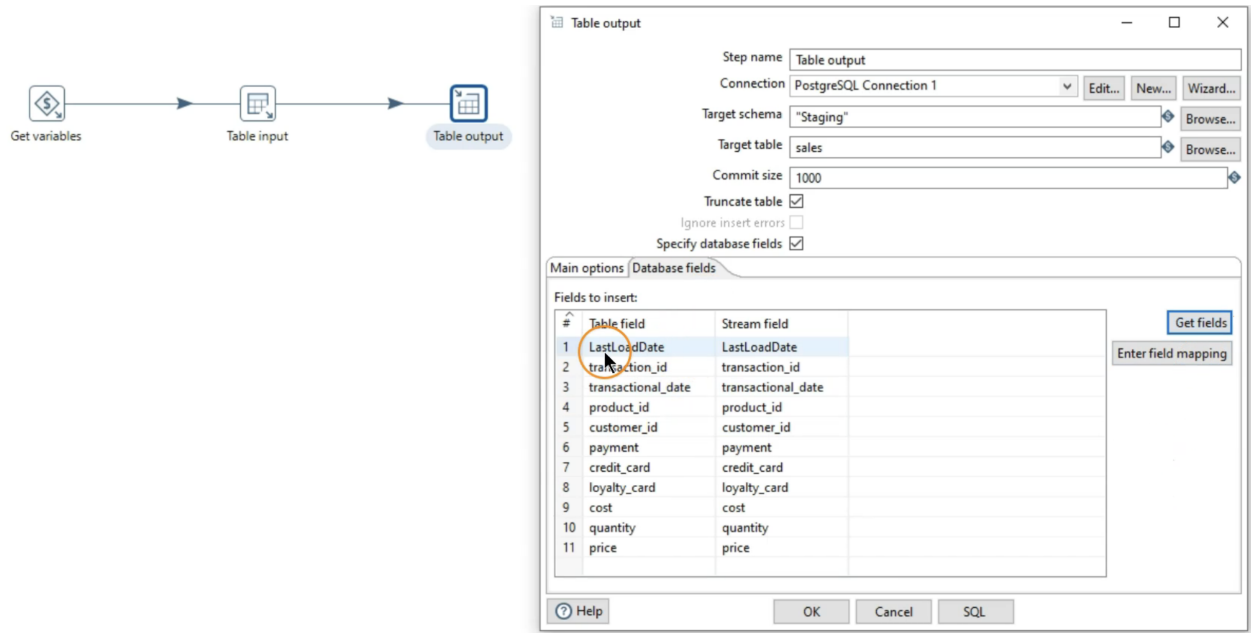


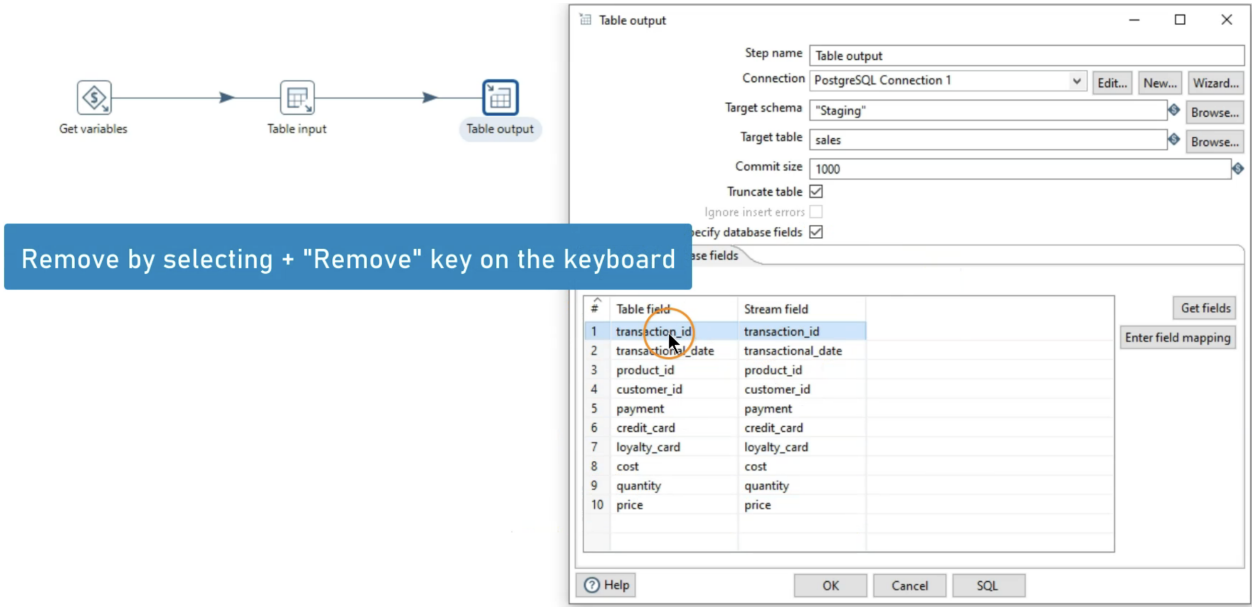




- There is no data in our staging table

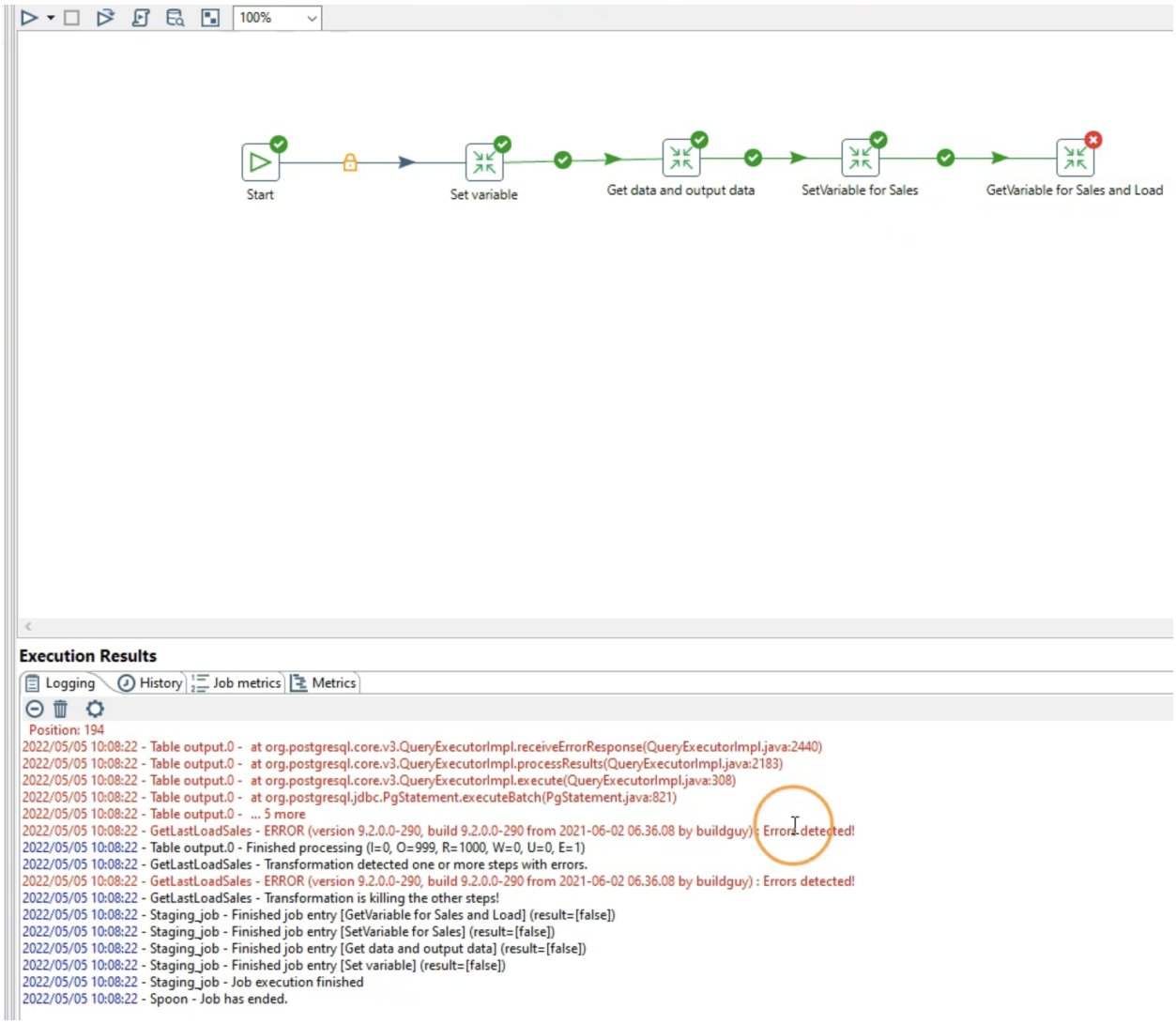



- Add TableOutput after getting the data from source. Remove the LastLoadDate from the GetFields

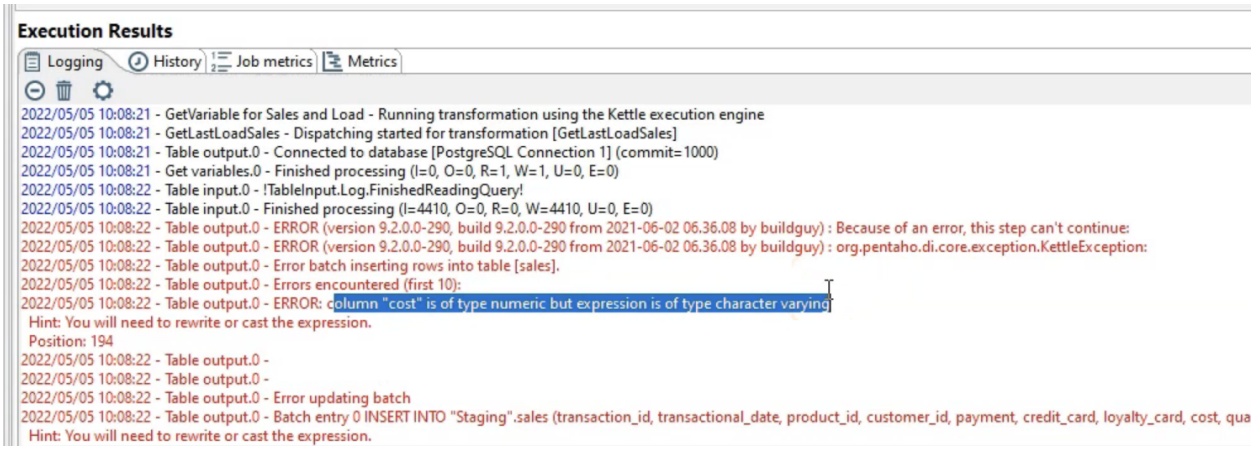


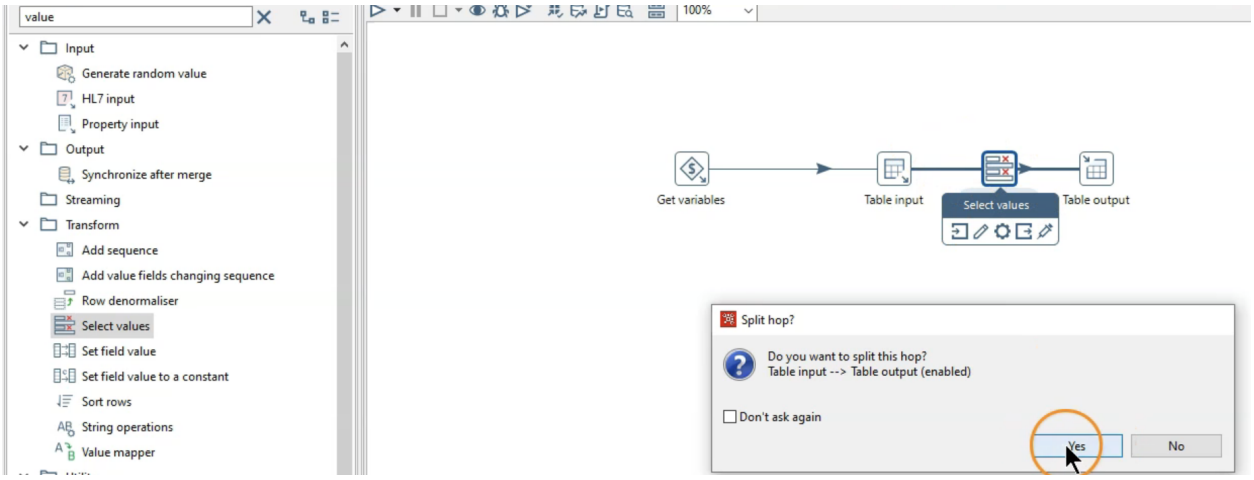

- Now save the transformation and run the Job again



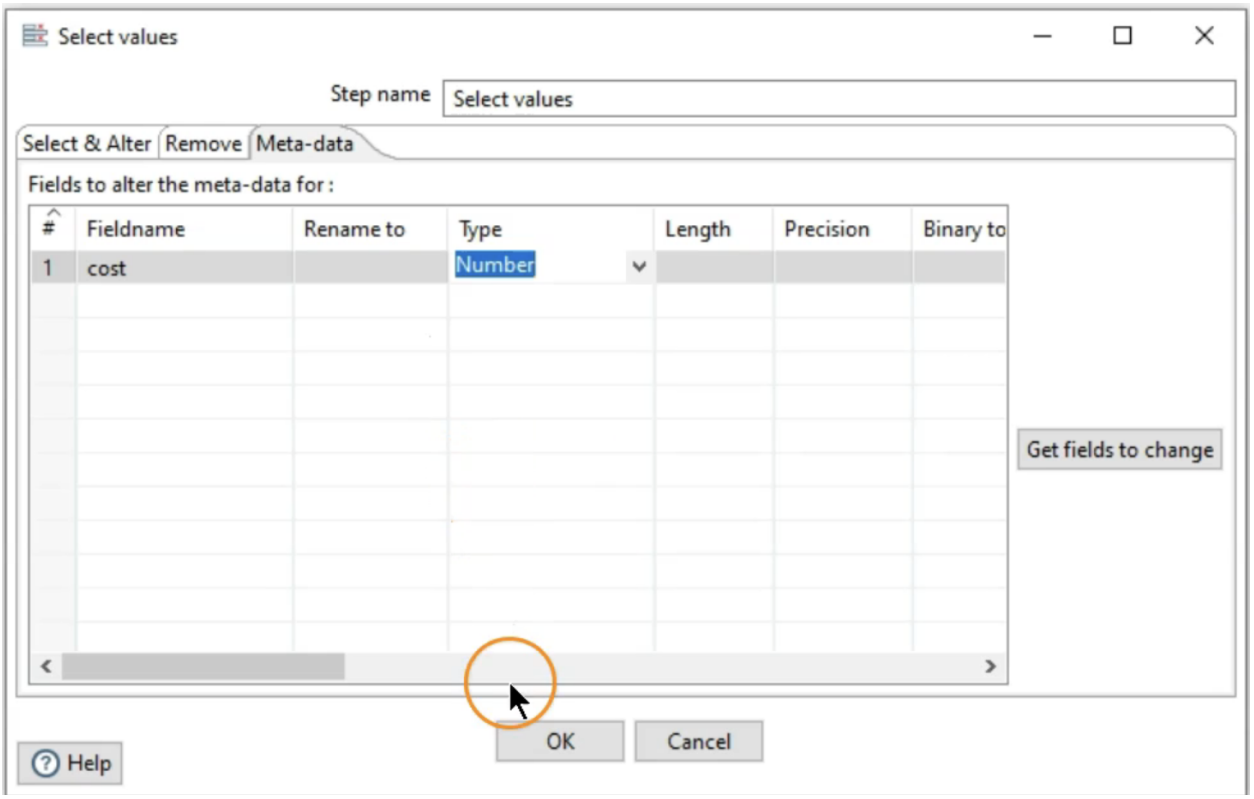

- We have some issues



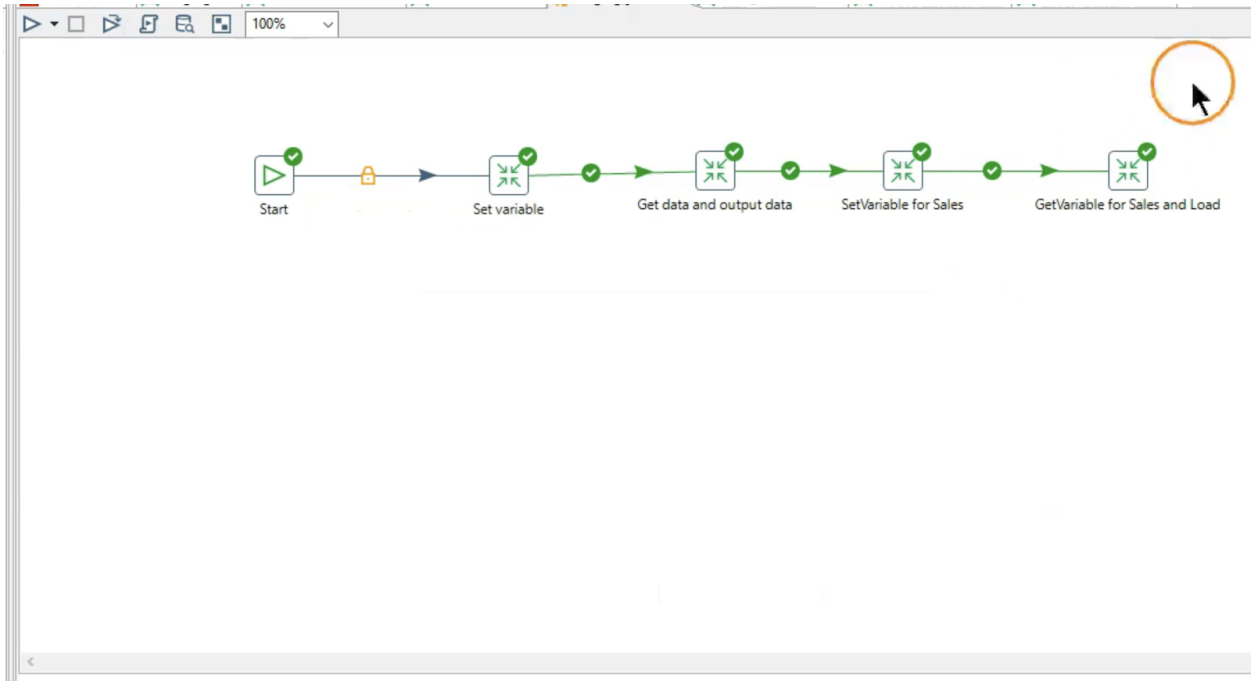
- Now to change the datatype, before the table output we need to make the data type change
- For this add ***Select Values*** before table output




- In meta-data we can change the data type of a column




- Now save and run the job again




**Execution Results**

Logging | History | Job metrics | Metrics

```
2022/05/05 10:11:01 - Set variables.0 - Finished processing (I=0, O=0, R=1, W=1, U=0, E=0)
2022/05/05 10:11:01 - Staging_job - Starting entry [GetVariable for Sales and Load]
2022/05/05 10:11:01 - GetVariable for Sales and Load - Using run configuration [Pentaho local]
2022/05/05 10:11:01 - GetVariable for Sales and Load - Running transformation using the Kettle execution engine
2022/05/05 10:11:01 - GetLastLoadSales - Dispatching started for transformation [GetLastLoadSales]
2022/05/05 10:11:01 - Table output.0 - Connected to database [PostgreSQL Connection 1] (commit=1000)
2022/05/05 10:11:01 - Get variables.0 - Finished processing (I=0, O=0, R=1, W=1, U=0, E=0)
2022/05/05 10:11:01 - Table input.0 - !TableInput.Log.FinishedReadingQuery!
2022/05/05 10:11:01 - Table input.0 - Finished processing (I=4410, O=0, R=0, W=4410, U=0, E=0)
2022/05/05 10:11:01 - Select values.0 - Finished processing (I=0, O=0, R=4410, W=4410, U=0, E=0)
2022/05/05 10:11:01 - Table output.0 - Finished processing (I=0, O=4410, R=4410, W=4410, U=0, E=0)
2022/05/05 10:11:01 - Staging_job - Finished job entry [GetVariable for Sales and Load] (result=[true])
2022/05/05 10:11:01 - Staging_job - Finished job entry [SetVariable for Sales] (result=[true])
2022/05/05 10:11:01 - Staging_job - Finished job entry [Get data and output data] (result=[true])
2022/05/05 10:11:01 - Staging_job - Finished job entry [Set variable] (result=[true])
2022/05/05 10:11:01 - Staging_job - Job execution finished
2022/05/05 10:11:01 - Spoon - Job has ended.
```

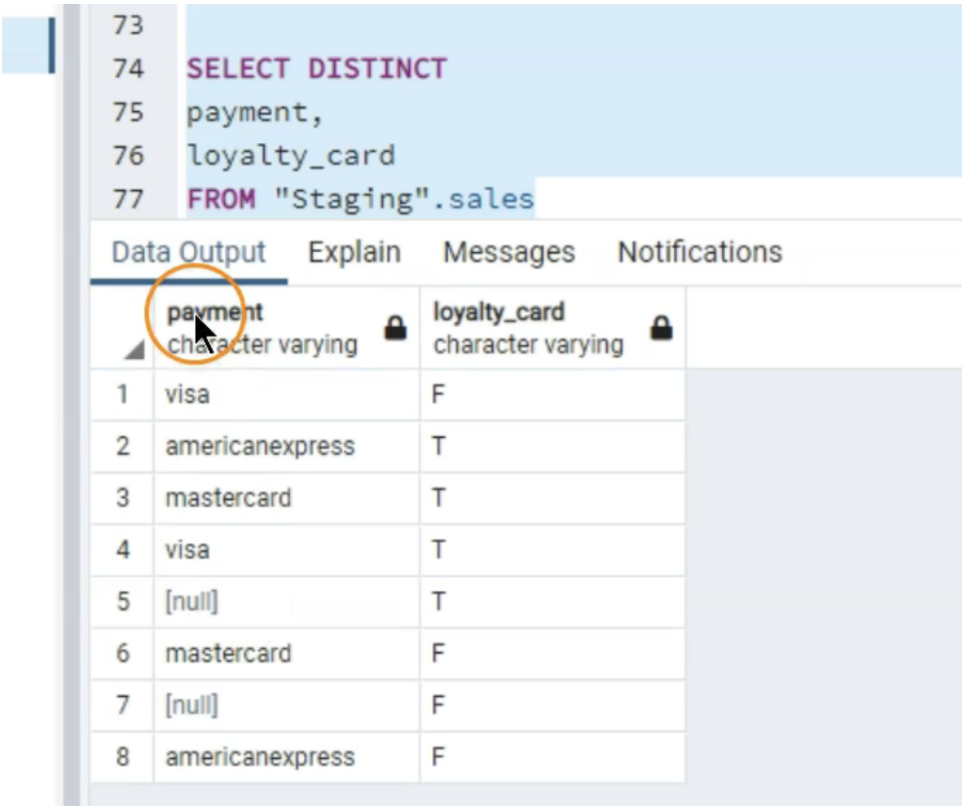
```
70
71  SELECT * FROM "Staging".sales;
72
73
74
```

Data Output | Explain | Messages | Notifications

| | transaction_id [PK] integer | transactional_date timestamp without time zone | product_id character varying | customer_id integer | payment character varying | credit_card bigint | loyalty_card character varying | cost numeric | quantity integer | price numeric |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2021-05-04 02:00:00 | P0494 | 4 | visa | 41593010498829 | F | 17.33 | 2 | 18.29 |
| 2 | 2 | 2021-05-04 03:04:00 | P0221 | 5 | visa | 41596151234556 | F | 0.59 | 1 | 1.49 |
| 3 | 3 | 2021-05-04 03:56:00 | P0625 | 5 | visa | 41594885335898 | F | 5.15 | 3 | 5.89 |
| 4 | 4 | 2021-05-04 05:20:00 | P0431 | 8 | mastercard | 5308753677552345 | F | 10.67 | 2 | 11.59 |
| 5 | 5 | 2021-05-04 05:45:00 | P0058 | 5 | mastercard | 5308752372298261 | T | 11.38 | 2 | 12.39 |
| 6 | 6 | 2021-05-04 06:58:00 | P0385 | 6 | americanexpress | 374288563442549 | F | 13.22 | 1 | 14.69 |
| 7 | 7 | 2021-05-04 07:03:00 | P0575 | 4 | visa | 4041598869758 | F | 2.81 | 1 | 3.99 |
| 8 | 8 | 2021-05-04 07:45:00 | P0187 | 5 | americanexpress | 374283491107967 | F | 4.17 | 1 | 4.89 |
| 9 | 9 | 2021-05-04 09:58:00 | P0074 | 7 | mastercard | 5308753211196286 | F | 18.11 | 1 | 19.79 |
| 10 | 10 | 2021-05-04 15:51:00 | P0456 | 5 | mastercard | 5488370523083285 | T | 18.35 | 2 | 20.19 |
| 11 | 11 | 2021-05-04 16:13:00 | P0519 | 5 | mastercard | 5488379752512880 | F | 12.45 | 3 | 14.09 |
| 12 | 12 | 2021-05-04 17:07:00 | P0071 | 12 | visa | 41594984149257 | F | 13.09 | 5 | 14.69 |
| 13 | 13 | 2021-05-04 19:57:00 | P0550 | 5 | americanexpress | 374288299677079 | F | 15.35 | 1 | 17.09 |

# Load payment dimension

- Select only required columns from the staging table to identify the required number of combinations



```
73
74  SELECT DISTINCT
75  payment,
76  loyalty_card
77  FROM "Staging".sales
```

Data Output | Explain | Messages | Notifications

| | payment character varying | loyalty_card character varying |
|---|---|---|
| 1 | visa | F |
| 2 | americanexpress | T |
| 3 | mastercard | T |
| 4 | visa | T |
| 5 | [null] | T |
| 6 | mastercard | F |
| 7 | [null] | F |
| 8 | americanexpress | F |

- Here nulls in payment means there were not using any card means they were using Cash

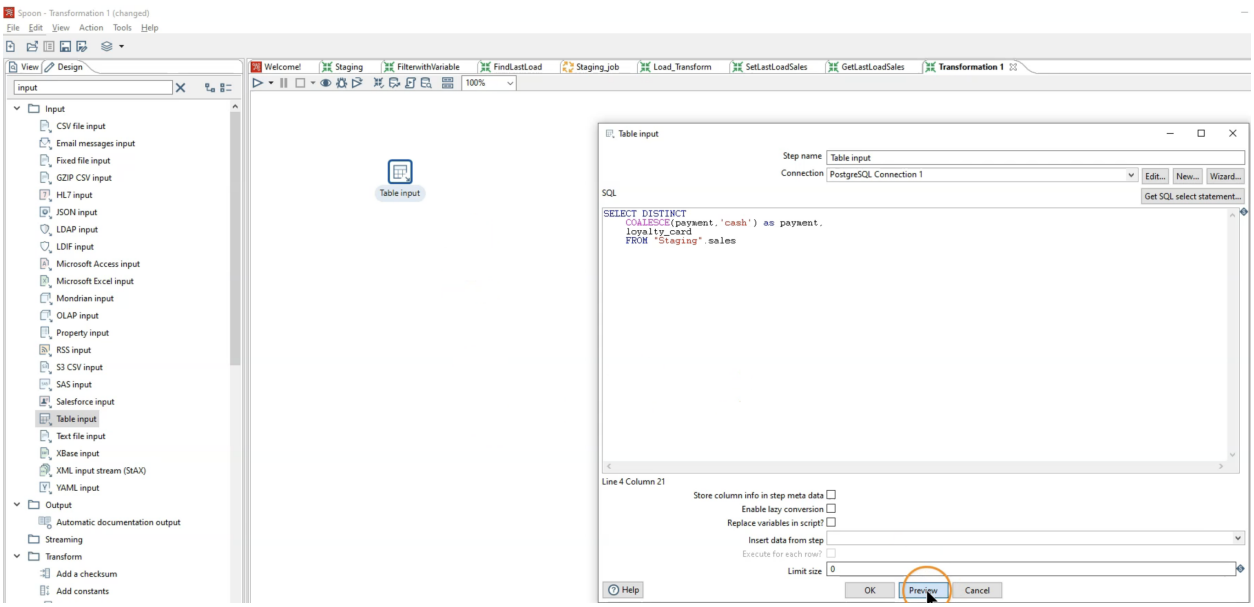
```sql
SELECT DISTINCT
COALESCE(payment,'cash') as payment,
loyalty_card
FROM "Staging".sales
```

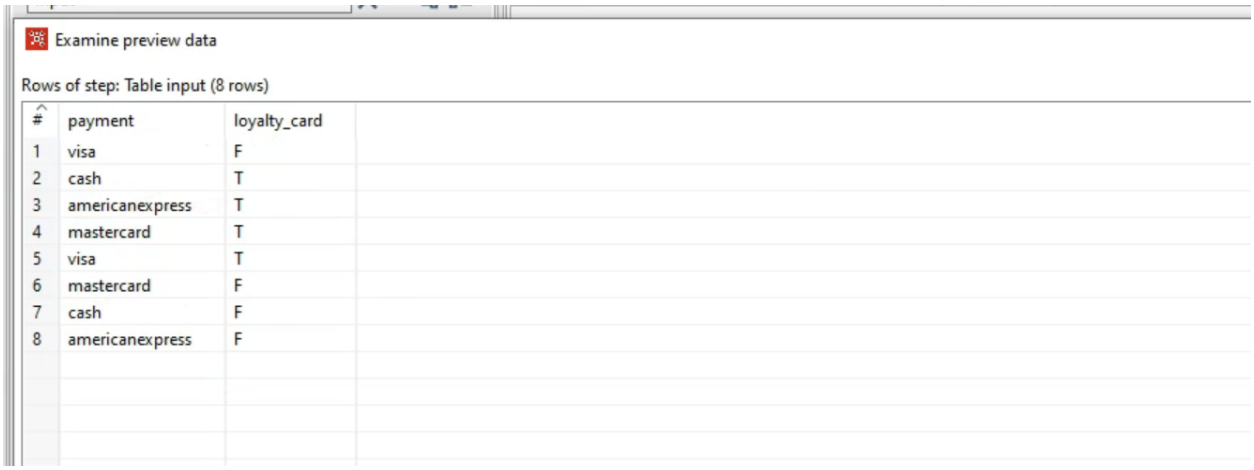
Data Output | Explain | Messages | Notifications

| | payment character varying | loyalty_card character varying |
|---|---|---|
| 1 | visa | F |
| 2 | cash | T |
| 3 | americanexpress | T |
| 4 | mastercard | T |
| 5 | visa | T |
| 6 | mastercard | F |
| 7 | cash | F |
| 8 | americanexpress | F |

```sql
CREATE TABLE core.dim_payment
(
    payment_pk integer NOT NULL GENERATED BY DEFAULT AS IDENTITY ( INCREMENT 1 START 1 ),
    payment character varying,
    loyalty_card character varying,
    PRIMARY KEY (payment_pk)
);
```

- Now let’s create new transformation to create dim_payment




| # | payment | loyalty_card |
|---|---|---|
| 1 | visa | F |
| 2 | cash | T |
| 3 | americanexpress | T |
| 4 | mastercard | T |
| 5 | visa | T |
| 6 | mastercard | F |
| 7 | cash | F |
| 8 | americanexpress | F |

- This is now a full load, but if there are new updates/inserts we need to take it

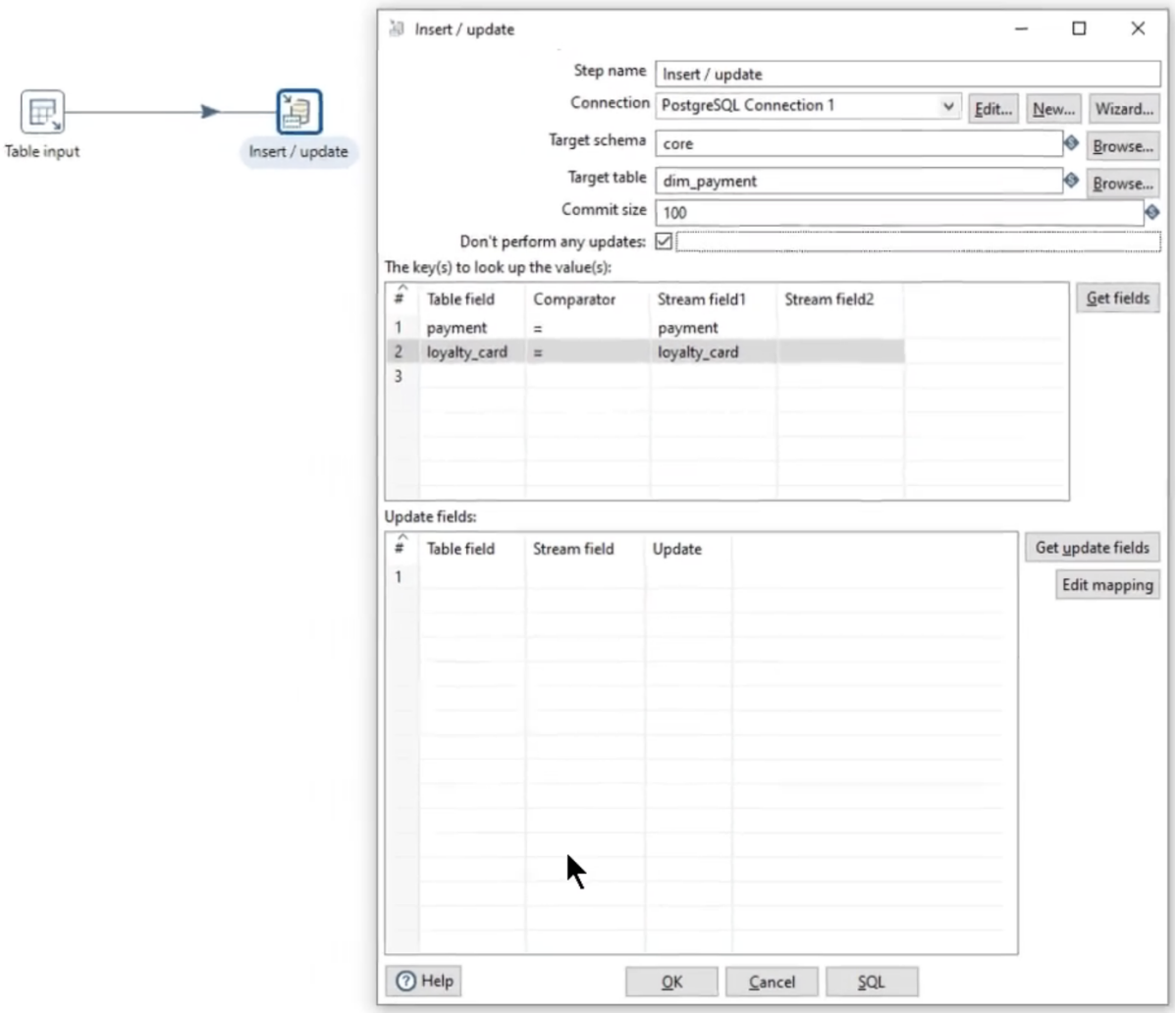
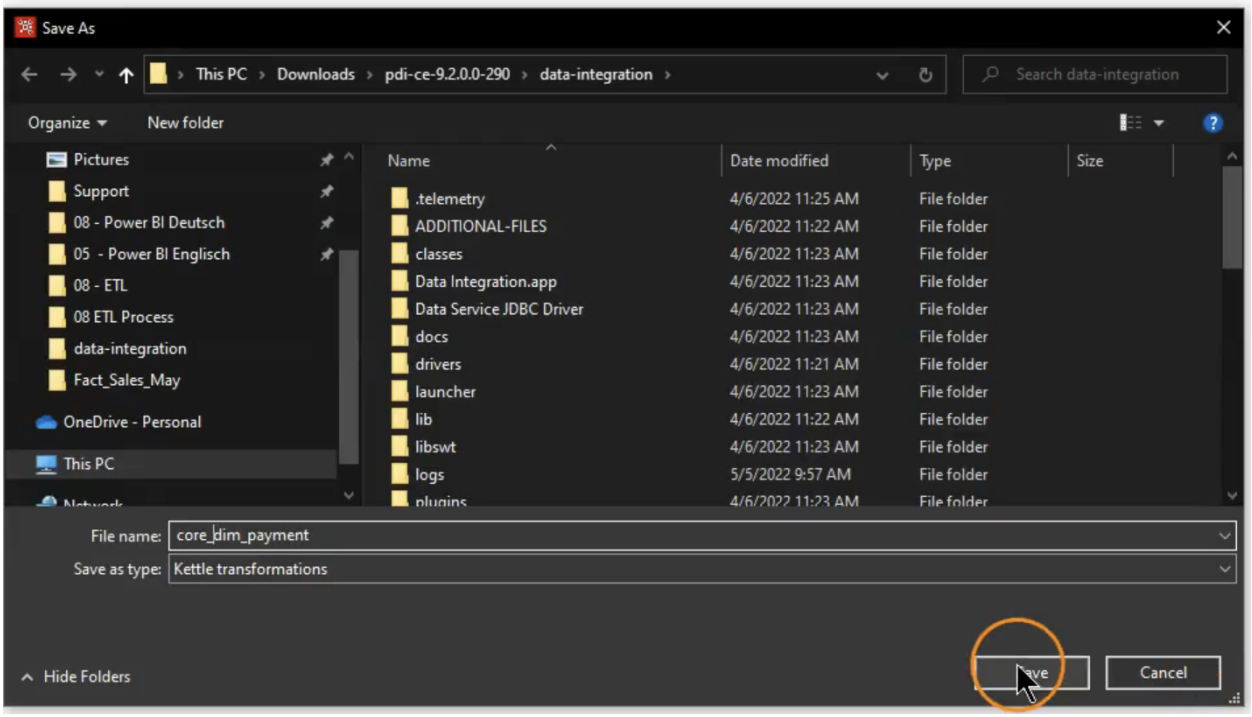


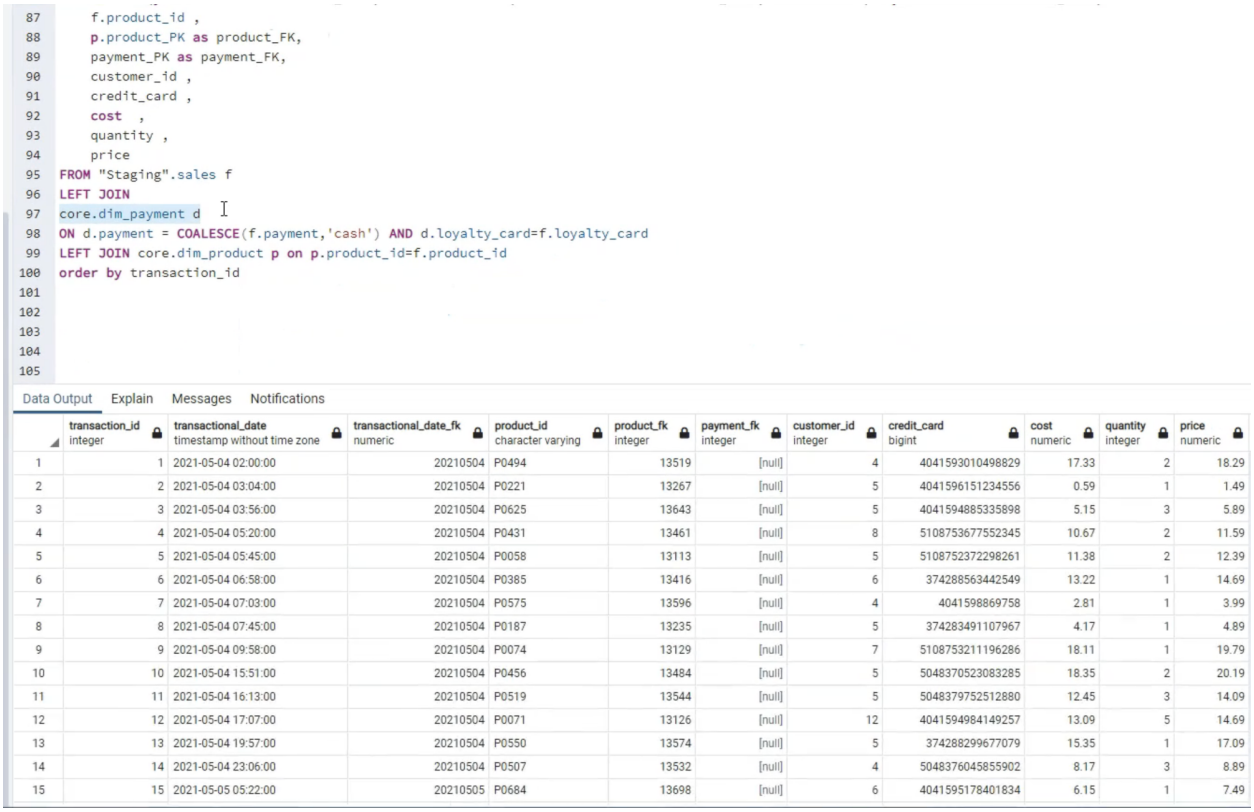
Table input
Insert / update
Insert / update
Step name
Insert / update
Connection
PostgreSQL Connection 1
Edit...
New...
Wizard...
Target schema
core
Browse...
Target table
dim_payment
Browse...
Commit size
100
Don't perform any updates:
The key(s) to look up the value(s):
# Table field Comparator Stream field1 Stream field2
1 payment = payment
2 loyalty_card = loyalty_card
3
Get fields
Update fields:
# Table field Stream field Update
1
Get update fields
Edit mapping
Help
OK
Cancel
SQL




# Transform and Load Sales Fact

```
SELECT
    transaction_id ,
    transactional_date ,
    EXTRACT(year from transactional_date)*10000 + EXTRACT('mo
nth' from transactional_date)*100+EXTRACT('day' from transact
ional_date)as   transactional_date_fk,
    f.product_id ,
    p.product_PK as product_FK,
    payment_PK as payment_FK,
    customer_id ,
    credit_card ,
    cost  ,
    quantity ,
    price
FROM "Staging".sales f
LEFT JOIN
core.dim_payment d
```

```
ON d.payment = COALESCE(f.payment,'cash') AND d.loyalty_card=
f.loyalty_card
LEFT JOIN core.dim_product p on p.product_id=f.product_id
order by transaction_id
```

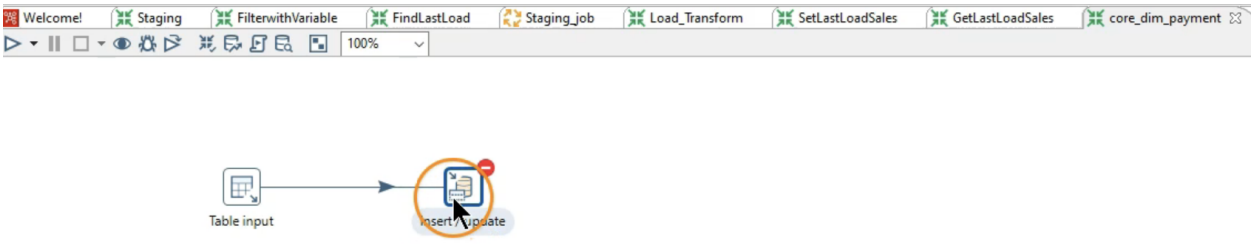


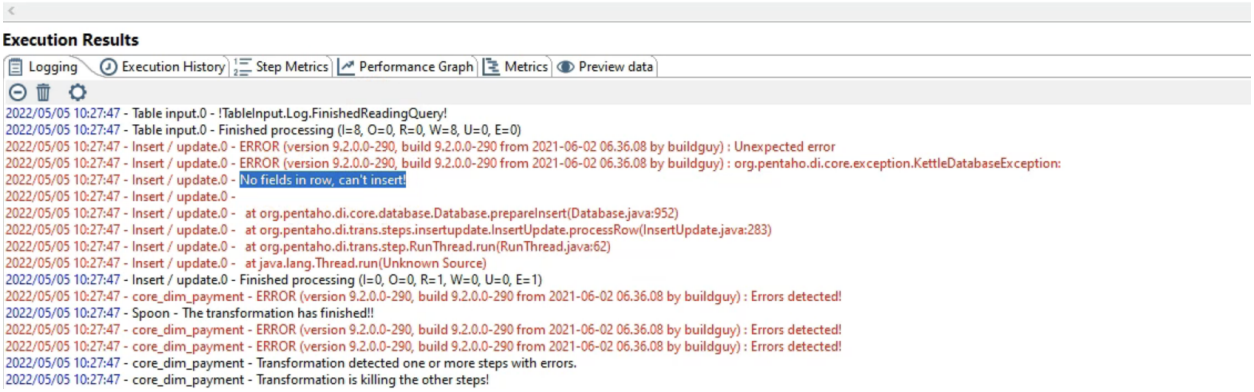
```
    f.product_id ,
    p.product_PK as product_FK,
    payment_PK as payment_FK,
    customer_id ,
    credit_card ,
    cost  ,
    quantity ,
    price
FROM "Staging".sales f
LEFT JOIN
core.dim_payment d
ON d.payment = COALESCE(f.payment,'cash') AND d.loyalty_card=f.loyalty_card
LEFT JOIN core.dim_product p on p.product_id=f.product_id
order by transaction_id
```

Data Output | Explain | Messages | Notifications

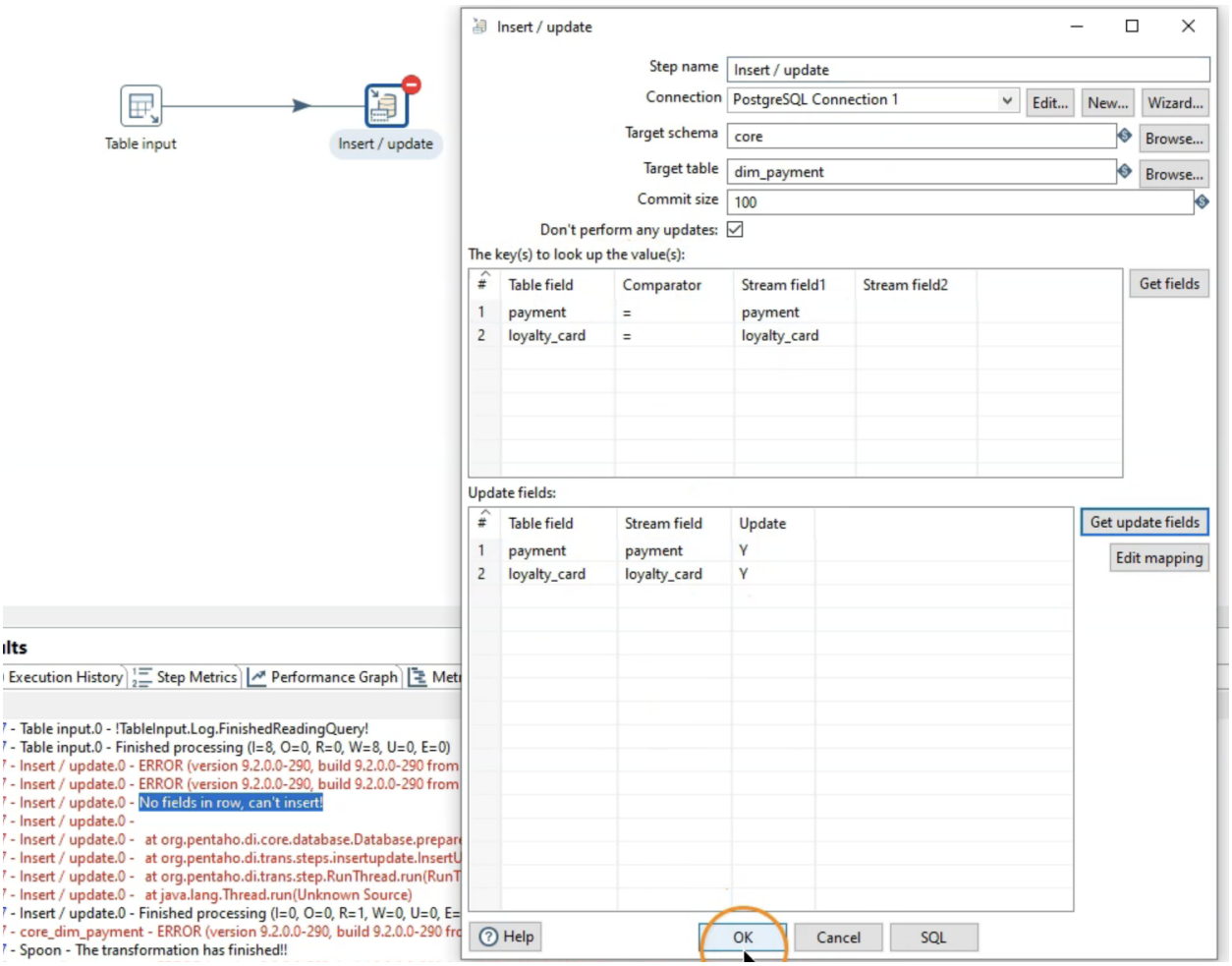
| | transaction_id integer | transactional_date timestamp without time zone | transactional_date_fk numeric | product_id character varying | product_fk integer | payment_fk integer | customer_id integer | credit_card bigint | cost numeric | quantity integer | price numeric |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2021-05-04 02:00:00 | 20210504 | P0494 | 13519 | [null] | 4 | 4041593010498829 | 17.33 | 2 | 18.29 |
| 2 | 2 | 2021-05-04 03:04:00 | 20210504 | P0221 | 13267 | [null] | 5 | 4041596151234556 | 0.59 | 1 | 1.49 |
| 3 | 3 | 2021-05-04 03:56:00 | 20210504 | P0625 | 13643 | [null] | 5 | 4041594885335898 | 5.15 | 3 | 5.89 |
| 4 | 4 | 2021-05-04 05:20:00 | 20210504 | P0431 | 13461 | [null] | 8 | 5108753677552345 | 10.67 | 2 | 11.59 |
| 5 | 5 | 2021-05-04 05:45:00 | 20210504 | P0058 | 13113 | [null] | 5 | 5108752372298261 | 11.38 | 2 | 12.39 |
| 6 | 6 | 2021-05-04 06:58:00 | 20210504 | P0385 | 13416 | [null] | 6 | 374288563442549 | 13.22 | 1 | 14.69 |
| 7 | 7 | 2021-05-04 07:03:00 | 20210504 | P0575 | 13596 | [null] | 4 | 4041598869758 | 2.81 | 1 | 3.99 |
| 8 | 8 | 2021-05-04 07:45:00 | 20210504 | P0187 | 13235 | [null] | 5 | 374283491107967 | 4.17 | 1 | 4.89 |
| 9 | 9 | 2021-05-04 09:58:00 | 20210504 | P0074 | 13129 | [null] | 7 | 5108753211196286 | 18.11 | 1 | 19.79 |
| 10 | 10 | 2021-05-04 15:51:00 | 20210504 | P0456 | 13484 | [null] | 5 | 5048370523083285 | 18.35 | 2 | 20.19 |
| 11 | 11 | 2021-05-04 16:13:00 | 20210504 | P0519 | 13544 | [null] | 5 | 5048379752512880 | 12.45 | 3 | 14.09 |
| 12 | 12 | 2021-05-04 17:07:00 | 20210504 | P0071 | 13126 | [null] | 12 | 4041594984149257 | 13.09 | 5 | 14.69 |
| 13 | 13 | 2021-05-04 19:57:00 | 20210504 | P0550 | 13574 | [null] | 5 | 374288299677079 | 15.35 | 1 | 17.09 |
| 14 | 14 | 2021-05-04 23:06:00 | 20210504 | P0507 | 13532 | [null] | 4 | 504837604585902 | 8.17 | 3 | 8.89 |
| 15 | 15 | 2021-05-05 05:22:00 | 20210505 | P0684 | 13698 | [null] | 6 | 40415951784018 34 | 6.15 | 1 | 7.49 |

- We do not have any values in payment_fk, we have only set up ETL but have not executed it




- We need to get the update fields by clicking get update Fields




- We need to set the Update to N for all fields

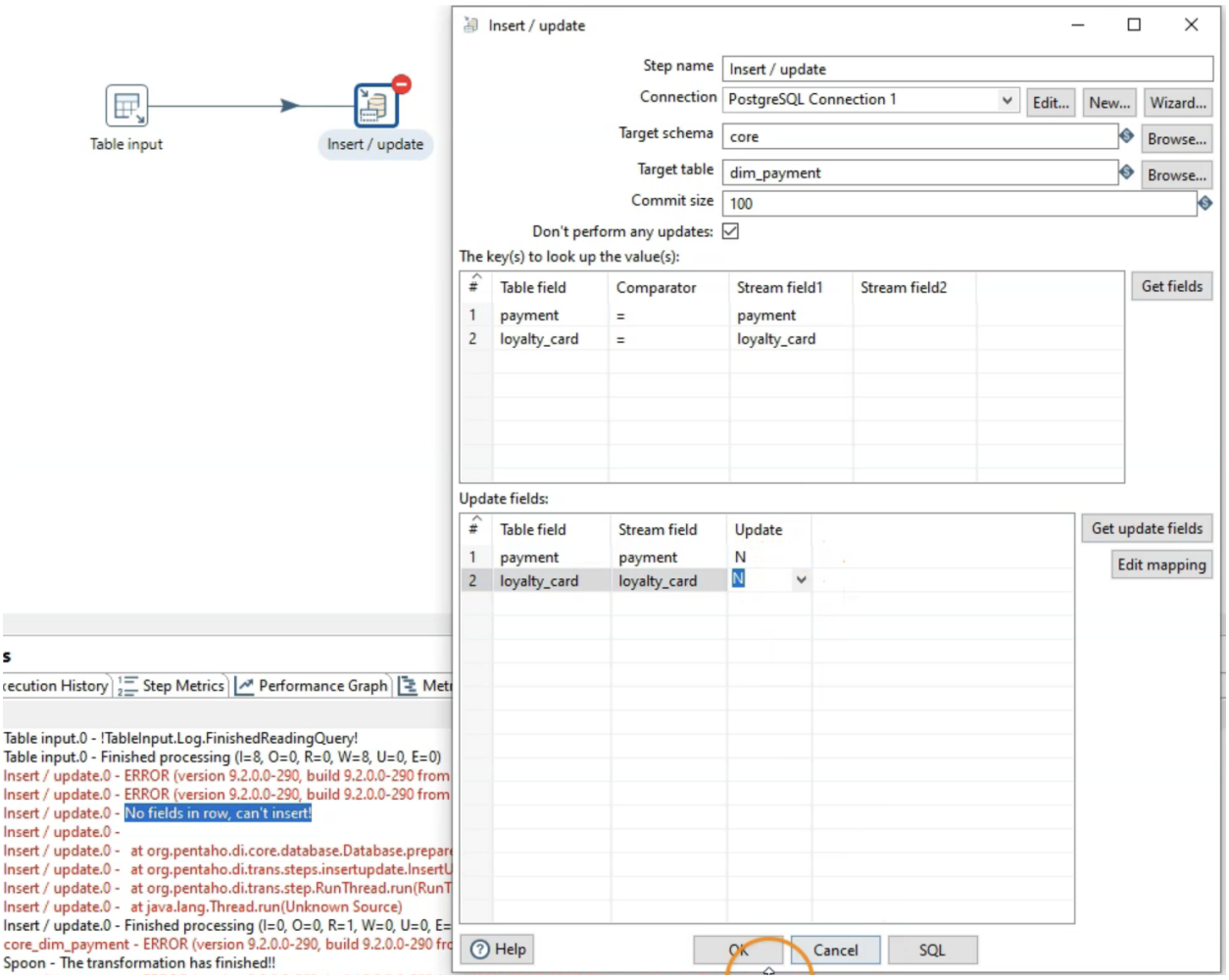
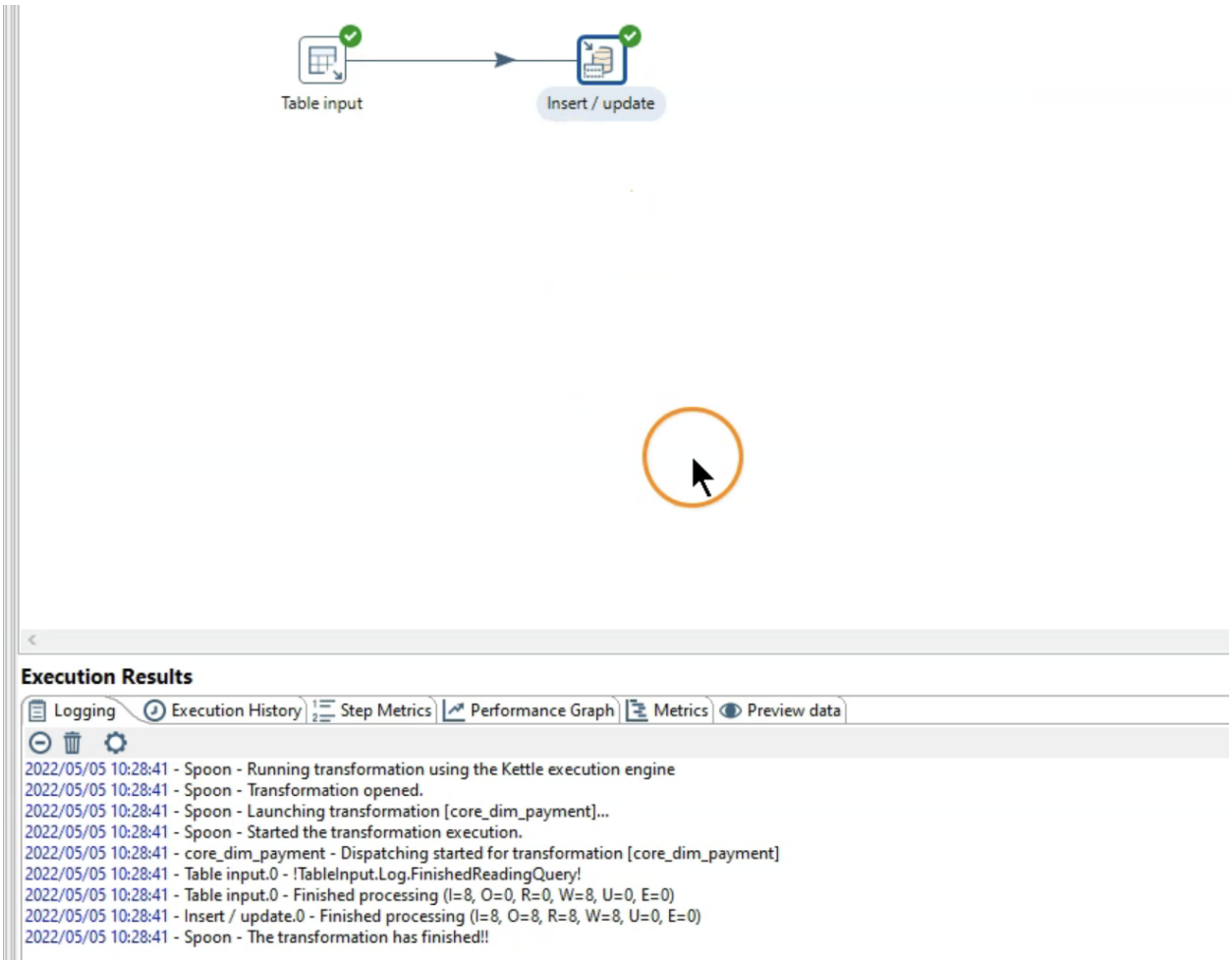


Table input
Insert / update
Insert / update
Step name
Insert / update
Connection
PostgreSQL Connection 1
Edit...
New...
Wizard...
Target schema
core
Browse...
Target table
dim_payment
Browse...
Commit size
100
Don't perform any updates:
The key(s) to look up the value(s):
# Table field Comparator Stream field1 Stream field2
1 payment = payment
2 loyalty_card = loyalty_card
Get fields
Update fields:
# Table field Stream field Update
1 payment payment N
2 loyalty_card loyalty_card N
Get update fields
Edit mapping
Help
OK
Cancel
SQL
Step Metrics
Performance Graph
Table input.0 - !TableInput.Log.FinishedReadingQuery!
Table input.0 - Finished processing (I=8, O=0, R=0, W=8, U=0, E=0)
Insert / update.0 - ERROR (version 9.2.0.0-290, build 9.2.0.0-290 from
Insert / update.0 - ERROR (version 9.2.0.0-290, build 9.2.0.0-290 from
Insert / update.0 - No fields in row, can't insert!
Insert / update.0 -
Insert / update.0 - at org.pentaho.di.core.database.Database.prepar
Insert / update.0 - at org.pentaho.di.trans.steps.insertupdate.InsertU
Insert / update.0 - at org.pentaho.di.trans.step.RunThread.run(RunT
Insert / update.0 - at java.lang.Thread.run(Unknown Source)
Insert / update.0 - Finished processing (I=0, O=0, R=1, W=0, U=0, E=
core_dim_payment - ERROR (version 9.2.0.0-290, build 9.2.0.0-290 fro
Spoon - The transformation has finished!!




- Since we now have data in the payment dimension table we should now get values in payment_fk

```
    f.product_id ,
    p.product_PK as product_FK,
    payment_PK as payment_FK,
    customer_id ,
    credit_card ,
    cost  ,
    quantity ,
    price
FROM "Staging".sales f
LEFT JOIN
core.dim_payment d
ON d.payment = COALESCE(f.payment,'cash') AND d.loyalty_card=f.loyalty_card
LEFT JOIN core.dim_product p on p.product_id=f.product_id
order by transaction_id
```

Data Output | Explain | Messages | Notifications

| | transaction_id integer | transactional_date timestamp without time zone | transactional_date_fk numeric | product_id character varying | product_fk integer | payment_fk integer | customer_id integer | credit_card bigint | cost numeric | quantity integer | price numeric |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2021-05-04 02:00:00 | 20210504 | P0494 | 13519 | 1 | 4 | 4041593010498829 | 17.33 | 2 | 18.29 |
| 2 | 2 | 2021-05-04 03:04:00 | 20210504 | P0221 | 13267 | 1 | 5 | 4041596151234556 | 0.59 | 1 | 1.49 |
| 3 | 3 | 2021-05-04 03:56:00 | 20210504 | P0625 | 13643 | 1 | 5 | 4041594885335898 | 5.15 | 3 | 5.89 |
| 4 | 4 | 2021-05-04 05:20:00 | 20210504 | P0431 | 13461 | 6 | 8 | 5108753677552345 | 10.67 | 2 | 11.59 |
| 5 | 5 | 2021-05-04 05:45:00 | 20210504 | P0058 | 13113 | 4 | 5 | 5108752372298261 | 11.38 | 2 | 12.39 |
| 6 | 6 | 2021-05-04 06:58:00 | 20210504 | P0385 | 13416 | 8 | 6 | 374288563442549 | 13.22 | 1 | 14.69 |
| 7 | 7 | 2021-05-04 07:03:00 | 20210504 | P0575 | 13596 | 1 | 4 | 4041598869758 | 2.81 | 1 | 3.99 |
| 8 | 8 | 2021-05-04 07:45:00 | 20210504 | P0187 | 13235 | 8 | 5 | 374283491107967 | 4.17 | 1 | 4.89 |
| 9 | 9 | 2021-05-04 09:58:00 | 20210504 | P0074 | 13129 | 6 | 7 | 5108753211196286 | 18.11 | 1 | 19.79 |
| 10 | 10 | 2021-05-04 15:51:00 | 20210504 | P0456 | 13484 | 4 | 5 | 5048370523083285 | 18.35 | 2 | 20.19 |
| 11 | 11 | 2021-05-04 16:13:00 | 20210504 | P0519 | 13544 | 6 | 5 | 5048379752512880 | 12.45 | 3 | 14.09 |
| 12 | 12 | 2021-05-04 17:07:00 | 20210504 | P0071 | 13126 | 1 | 12 | 4041594984149257 | 13.09 | 5 | 14.69 |
| 13 | 13 | 2021-05-04 19:57:00 | 20210504 | P0550 | 13574 | 8 | 5 | 374288299677079 | 15.35 | 1 | 17.09 |

- Extract the transactional_date_fk

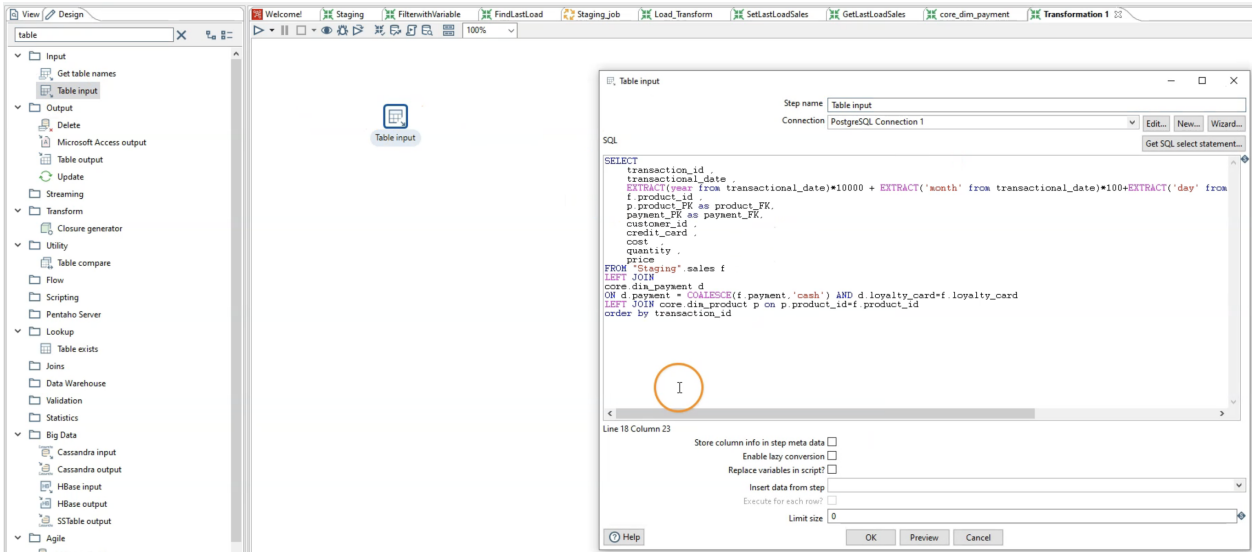
```
SELECT
    transaction_id ,
    transactional_date ,
    EXTRACT(year from transactional_date)*10000 + EXTRACT('month' from transactional_date)*100+EXTRACT('day' from transactional_date)as     transactional_date_fk,
    f.product_id ,
    p.product_PK as prod
    payment_PK as paymen
    customer_id ,
    credit_card ,
    cost  ,
    quantity ,
    price
FROM "Staging".sales f
LEFT JOIN
core.dim_payment d
ON d.payment = COALESCE(f.payment,'cash') AND d.loyalty_card=f.loyalty_card
LEFT JOIN core.dim_product p on p.product_id=f.product_id
```

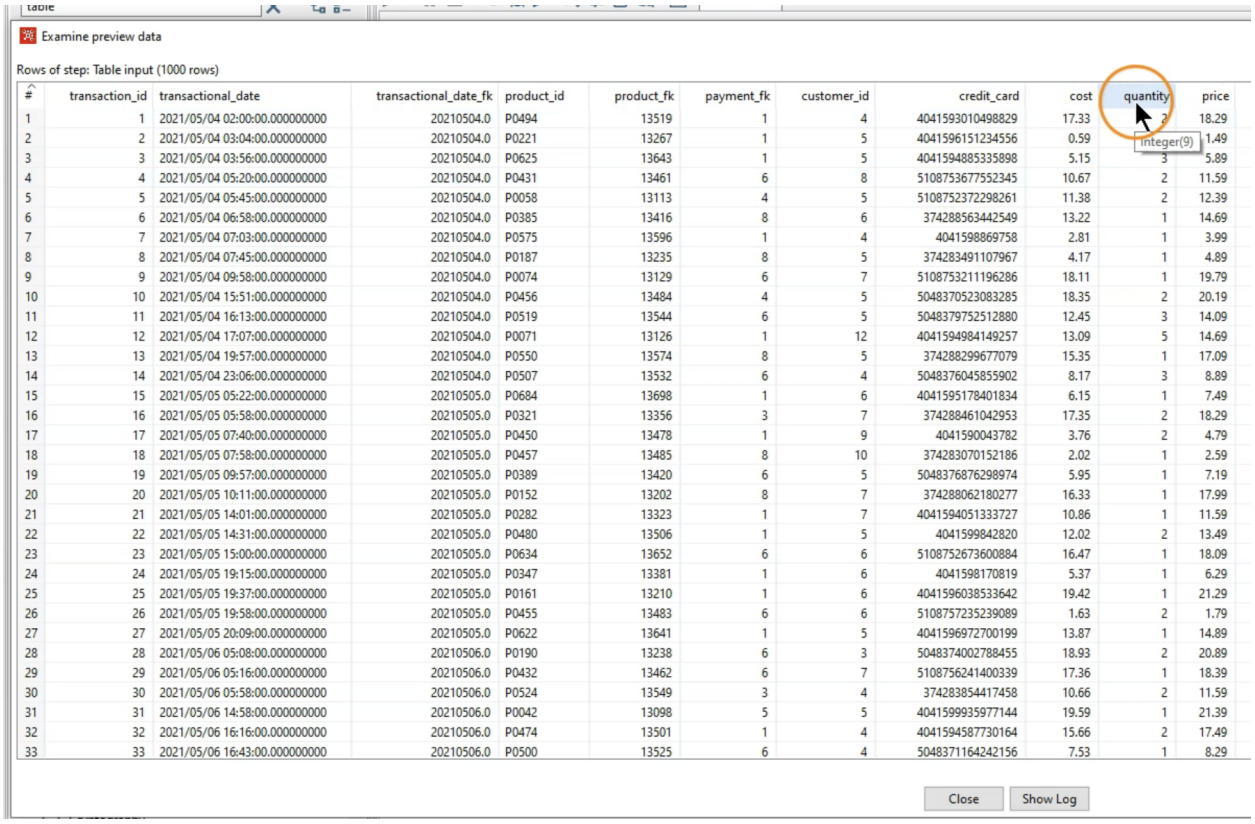
20210000 20210500 20210504

ta Output | Explain | Messages | Notifications

| | transaction_id integer | transactional_date timestamp without time zone | transactional_date_fk numeric | product_id character varying | product_fk integer | payment_fk integer | customer_id integer | credit_card bigint | cost numeric | quantity integer | price numeric |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2021-05-04 02:00:00 | 20210504 | P0494 | 13519 | 1 | 4 | 4041593010498829 | 17.33 | 2 | 18.29 |
| | 2 | 2021-05-04 03:04:00 | 20210504 | P0221 | 13267 | 1 | 5 | 4041596151234556 | 0.59 | 1 | 1.49 |
| | 3 | 2021-05-04 03:56:00 | 20210504 | P0625 | 13643 | 1 | 5 | 4041594885335898 | 5.15 | 3 | 5.89 |
| | 4 | 2021-05-04 05:20:00 | 20210504 | P0431 | 13461 | 6 | 8 | 5108753677552345 | 10.67 | 2 | 11.59 |
| | 5 | 2021-05-04 05:45:00 | 20210504 | P0058 | 13113 | 4 | 5 | 5108752372298261 | 11.38 | 2 | 12.39 |
| | 6 | 2021-05-04 06:58:00 | 20210504 | P0385 | 13416 | 8 | 6 | 374288563442549 | 13.22 | 1 | 14.69 |
| | 7 | 2021-05-04 07:03:00 | 20210504 | P0575 | 13596 | 1 | 4 | 4041598869758 | 2.81 | 1 | 3.99 |

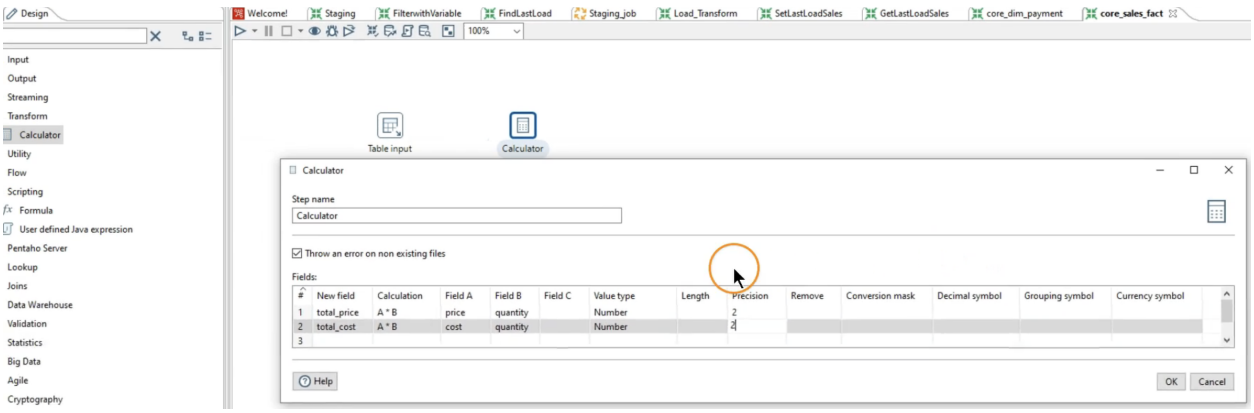
- Now set up new transformation and preview the data




| # | transaction_id | transactional_date | transactional_date_fk | product_id | product_fk | payment_fk | customer_id | credit_card | cost | quantity | price |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2021/05/04 02:00:00.000000000 | 20210504.0 | P0494 | 13519 | 1 | 4 | 4041593010498829 | 17.33 | 2 | 18.29 |
| 2 | 2 | 2021/05/04 03:04:00.000000000 | 20210504.0 | P0221 | 13267 | 1 | 5 | 4041596151234556 | 0.59 | 1 | 1.49 |
| 3 | 3 | 2021/05/04 03:56:00.000000000 | 20210504.0 | P0625 | 13643 | 1 | 5 | 4041594885335898 | 5.15 | 3 | 5.89 |
| 4 | 4 | 2021/05/04 05:20:00.000000000 | 20210504.0 | P0431 | 13461 | 6 | 8 | 5108753677552345 | 10.67 | 2 | 11.59 |
| 5 | 5 | 2021/05/04 05:45:00.000000000 | 20210504.0 | P0058 | 13113 | 4 | 5 | 5108752372298261 | 11.38 | 2 | 12.39 |
| 6 | 6 | 2021/05/04 06:58:00.000000000 | 20210504.0 | P0385 | 13416 | 8 | 6 | 374288563442549 | 13.22 | 1 | 14.69 |
| 7 | 7 | 2021/05/04 07:03:00.000000000 | 20210504.0 | P0575 | 13596 | 1 | 4 | 4041598869758 | 2.81 | 1 | 3.99 |
| 8 | 8 | 2021/05/04 07:45:00.000000000 | 20210504.0 | P0187 | 13235 | 8 | 5 | 374283491107967 | 4.17 | 1 | 4.89 |
| 9 | 9 | 2021/05/04 09:58:00.000000000 | 20210504.0 | P0074 | 13129 | 6 | 7 | 5108753211196286 | 18.11 | 1 | 19.79 |
| 10 | 10 | 2021/05/04 15:51:00.000000000 | 20210504.0 | P0456 | 13484 | 4 | 5 | 5048370523083285 | 18.35 | 2 | 20.19 |
| 11 | 11 | 2021/05/04 16:13:00.000000000 | 20210504.0 | P0519 | 13544 | 6 | 5 | 5048379752512880 | 12.45 | 3 | 14.09 |
| 12 | 12 | 2021/05/04 17:07:00.000000000 | 20210504.0 | P0071 | 13126 | 1 | 12 | 4041594984149257 | 13.09 | 5 | 14.69 |
| 13 | 13 | 2021/05/04 19:57:00.000000000 | 20210504.0 | P0550 | 13574 | 8 | 5 | 374288299677079 | 15.35 | 1 | 17.09 |
| 14 | 14 | 2021/05/04 23:06:00.000000000 | 20210504.0 | P0507 | 13532 | 6 | 4 | 5048376045855902 | 8.17 | 3 | 8.89 |
| 15 | 15 | 2021/05/05 05:22:00.000000000 | 20210505.0 | P0684 | 13698 | 1 | 6 | 4041595178401834 | 6.15 | 1 | 7.49 |
| 16 | 16 | 2021/05/05 05:58:00.000000000 | 20210505.0 | P0321 | 13356 | 3 | 7 | 374288461042953 | 17.35 | 2 | 18.29 |
| 17 | 17 | 2021/05/05 07:40:00.000000000 | 20210505.0 | P0450 | 13478 | 1 | 9 | 4041590043782 | 3.76 | 2 | 4.79 |
| 18 | 18 | 2021/05/05 07:58:00.000000000 | 20210505.0 | P0457 | 13485 | 8 | 10 | 374283070152186 | 2.02 | 1 | 2.59 |
| 19 | 19 | 2021/05/05 09:57:00.000000000 | 20210505.0 | P0389 | 13420 | 6 | 5 | 5048376876298974 | 5.95 | 1 | 7.19 |
| 20 | 20 | 2021/05/05 10:11:00.000000000 | 20210505.0 | P0152 | 13202 | 8 | 7 | 374288062180277 | 16.33 | 1 | 17.99 |
| 21 | 21 | 2021/05/05 14:01:00.000000000 | 20210505.0 | P0282 | 13323 | 1 | 7 | 4041594051333727 | 10.86 | 1 | 11.59 |
| 22 | 22 | 2021/05/05 14:31:00.000000000 | 20210505.0 | P0480 | 13506 | 1 | 5 | 4041599842820 | 12.02 | 2 | 13.49 |
| 23 | 23 | 2021/05/05 15:00:00.000000000 | 20210505.0 | P0634 | 13652 | 6 | 6 | 5108752673600884 | 16.47 | 1 | 18.09 |
| 24 | 24 | 2021/05/05 19:15:00.000000000 | 20210505.0 | P0347 | 13381 | 1 | 6 | 4041598170819 | 5.37 | 1 | 6.29 |
| 25 | 25 | 2021/05/05 19:37:00.000000000 | 20210505.0 | P0161 | 13210 | 1 | 6 | 4041596038533642 | 19.42 | 1 | 21.29 |
| 26 | 26 | 2021/05/05 19:58:00.000000000 | 20210505.0 | P0455 | 13483 | 6 | 6 | 5108757235239089 | 1.63 | 2 | 1.79 |
| 27 | 27 | 2021/05/05 20:09:00.000000000 | 20210505.0 | P0622 | 13641 | 1 | 5 | 4041596972700199 | 13.87 | 1 | 14.89 |
| 28 | 28 | 2021/05/06 05:08:00.000000000 | 20210506.0 | P0190 | 13238 | 6 | 3 | 5048374002788455 | 18.93 | 2 | 20.89 |
| 29 | 29 | 2021/05/06 05:16:00.000000000 | 20210506.0 | P0432 | 13462 | 6 | 7 | 5108756241400339 | 17.36 | 1 | 18.39 |
| 30 | 30 | 2021/05/06 05:58:00.000000000 | 20210506.0 | P0524 | 13549 | 3 | 4 | 374283854417458 | 10.66 | 2 | 11.59 |
| 31 | 31 | 2021/05/06 14:58:00.000000000 | 20210506.0 | P0042 | 13098 | 5 | 5 | 4041599935977144 | 19.59 | 1 | 21.39 |
| 32 | 32 | 2021/05/06 16:16:00.000000000 | 20210506.0 | P0474 | 13501 | 1 | 4 | 4041594587730164 | 15.66 | 2 | 17.49 |
| 33 | 33 | 2021/05/06 16:43:00.000000000 | 20210506.0 | P0500 | 13525 | 6 | 4 | 5048371164242156 | 7.53 | 1 | 8.29 |

- For additional calculated fields we can do it in SQL or we can do in the ETL tool
- To perform some calculation like SUM, AVG, COUNT we use calculator



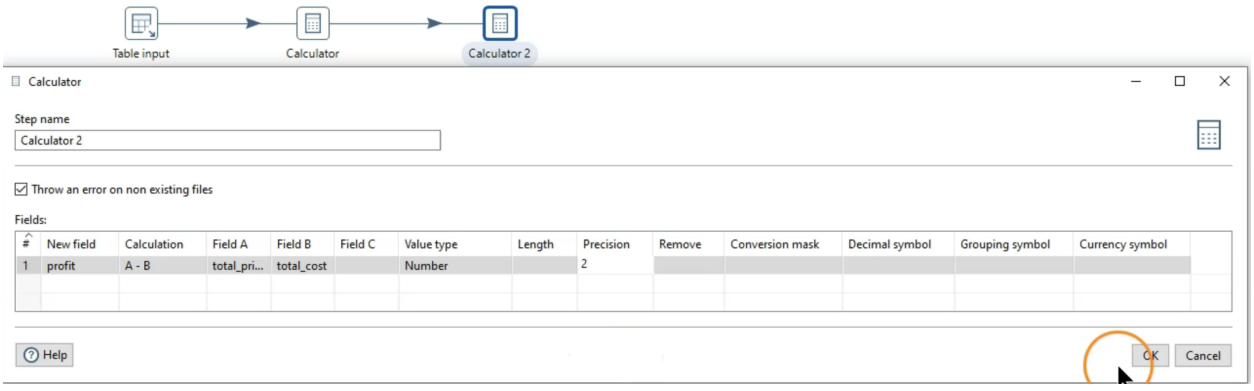

Examine preview data

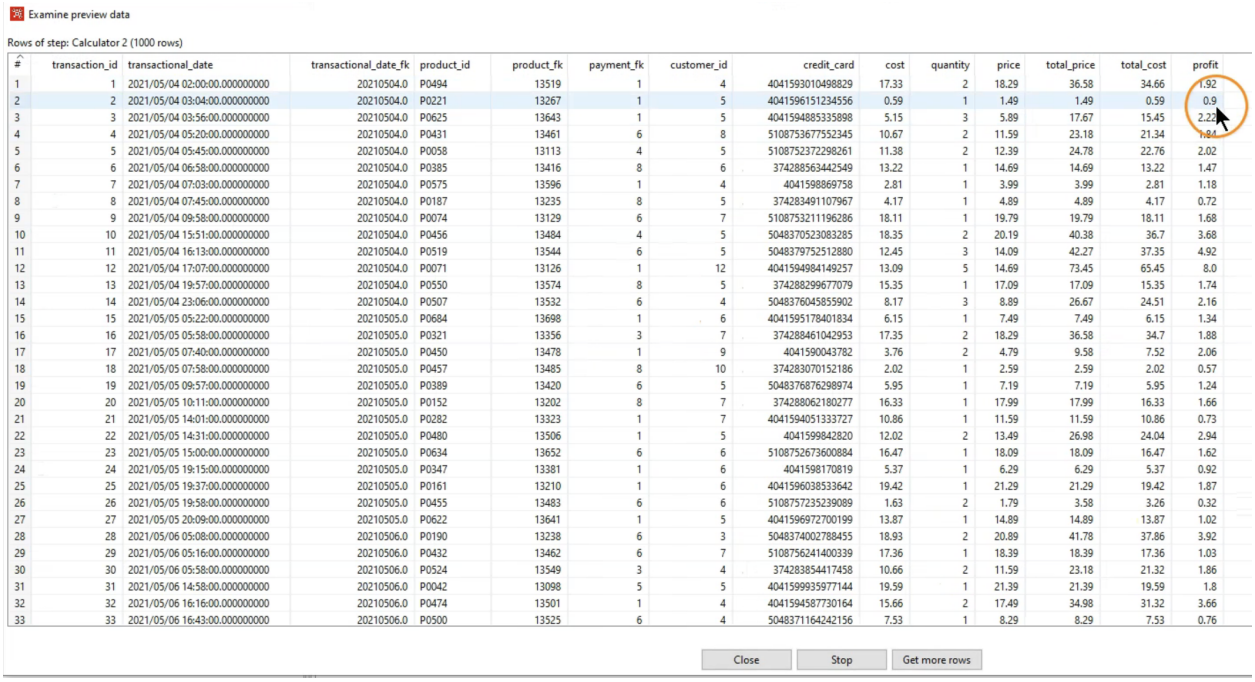
Rows of step: Calculator (1000 rows)

| # | transaction_id | transactional_date | transactional_date_fk | product_id | product_fk | payment_fk | customer_id | credit_card | cost | quantity | price | total_price | total_cost |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2021/05/04 02:00:00.000000000 | 20210504.0 | P0494 | 13519 | 1 | 4 | 4041593010498829 | 17.33 | 2 | 18.29 | 36.58 | 34.66 |
| 2 | 2 | 2021/05/04 03:04:00.000000000 | 20210504.0 | P0221 | 13267 | 1 | 5 | 4041596151234556 | 0.59 | 1 | 1.49 | 1.49 | 0.59 |
| 3 | 3 | 2021/05/04 03:56:00.000000000 | 20210504.0 | P0625 | 13643 | 1 | 5 | 4041594885335898 | 5.15 | 3 | 5.89 | 17.67 | 15.45 |
| 4 | 4 | 2021/05/04 05:20:00.000000000 | 20210504.0 | P0431 | 13461 | 6 | 8 | 5108753677552345 | 10.67 | 2 | 11.59 | 23.18 | 21.34 |
| 5 | 5 | 2021/05/04 05:45:00.000000000 | 20210504.0 | P0058 | 13113 | 4 | 5 | 5108752372298261 | 11.38 | 2 | 12.39 | 24.78 | 22.76 |
| 6 | 6 | 2021/05/04 06:58:00.000000000 | 20210504.0 | P0385 | 13416 | 8 | 6 | 374288563442549 | 13.22 | 1 | 14.69 | 14.69 | 13.22 |
| 7 | 7 | 2021/05/04 07:03:00.000000000 | 20210504.0 | P0575 | 13596 | 1 | 4 | 4041598869758 | 2.81 | 1 | 3.99 | 3.99 | 2.81 |
| 8 | 8 | 2021/05/04 07:45:00.000000000 | 20210504.0 | P0187 | 13235 | 8 | 5 | 374283491107967 | 4.17 | 1 | 4.89 | 4.89 | 4.17 |
| 9 | 9 | 2021/05/04 09:58:00.000000000 | 20210504.0 | P0074 | 13129 | 6 | 7 | 5108753211196286 | 18.11 | 1 | 19.79 | 19.79 | 18.11 |
| 10 | 10 | 2021/05/04 15:51:00.000000000 | 20210504.0 | P0456 | 13484 | 4 | 5 | 5048370523083285 | 18.35 | 2 | 20.19 | 40.38 | 36.7 |
| 11 | 11 | 2021/05/04 16:13:00.000000000 | 20210504.0 | P0519 | 13544 | 6 | 5 | 5048379752512880 | 12.45 | 3 | 14.09 | 42.27 | 37.35 |
| 12 | 12 | 2021/05/04 17:07:00.000000000 | 20210504.0 | P0071 | 13126 | 1 | 12 | 4041594984149257 | 13.09 | 5 | 14.69 | 73.45 | 65.45 |
| 13 | 13 | 2021/05/04 19:57:00.000000000 | 20210504.0 | P0550 | 13574 | 8 | 5 | 374288299677079 | 15.35 | 1 | 17.09 | 17.09 | 15.35 |
| 14 | 14 | 2021/05/04 23:06:00.000000000 | 20210504.0 | P0507 | 13532 | 6 | 4 | 5048376045855902 | 8.17 | 3 | 8.89 | 26.67 | 24.51 |
| 15 | 15 | 2021/05/05 05:22:00.000000000 | 20210505.0 | P0684 | 13698 | 1 | 6 | 4041595178401834 | 6.15 | 1 | 7.49 | 7.49 | 6.15 |
| 16 | 16 | 2021/05/05 05:58:00.000000000 | 20210505.0 | P0321 | 13356 | 3 | 7 | 374288461042953 | 17.35 | 2 | 18.29 | 36.58 | 34.7 |
| 17 | 17 | 2021/05/05 07:40:00.000000000 | 20210505.0 | P0450 | 13478 | 1 | 9 | 4041590043782 | 3.76 | 2 | 4.79 | 9.58 | 7.52 |
| 18 | 18 | 2021/05/05 07:58:00.000000000 | 20210505.0 | P0457 | 13485 | 8 | 10 | 374283070152186 | 2.02 | 1 | 2.59 | 2.59 | 2.02 |
| 19 | 19 | 2021/05/05 09:57:00.000000000 | 20210505.0 | P0389 | 13420 | 6 | 5 | 5048376876298974 | 5.95 | 1 | 7.19 | 7.19 | 5.95 |
| 20 | 20 | 2021/05/05 10:11:00.000000000 | 20210505.0 | P0152 | 13202 | 8 | 7 | 374288062180277 | 16.33 | 1 | 17.99 | 17.99 | 16.33 |
| 21 | 21 | 2021/05/05 14:01:00.000000000 | 20210505.0 | P0282 | 13323 | 1 | 7 | 4041594051333727 | 10.86 | 1 | 11.59 | 11.59 | 10.86 |
| 22 | 22 | 2021/05/05 14:31:00.000000000 | 20210505.0 | P0480 | 13506 | 1 | 5 | 4041599842820 | 12.02 | 2 | 13.49 | 26.98 | 24.04 |
| 23 | 23 | 2021/05/05 15:00:00.000000000 | 20210505.0 | P0634 | 13652 | 6 | 6 | 5108752673600884 | 16.47 | 1 | 18.09 | 18.09 | 16.47 |
| 24 | 24 | 2021/05/05 19:15:00.000000000 | 20210505.0 | P0347 | 13381 | 1 | 6 | 4041598170819 | 5.37 | 1 | 6.29 | 6.29 | 5.37 |
| 25 | 25 | 2021/05/05 19:37:00.000000000 | 20210505.0 | P0161 | 13210 | 1 | 6 | 4041596038533642 | 19.42 | 1 | 21.29 | 21.29 | 19.42 |
| 26 | 26 | 2021/05/05 19:58:00.000000000 | 20210505.0 | P0455 | 13483 | 6 | 6 | 5108757235239089 | 1.63 | 2 | 1.79 | 3.58 | 3.26 |
| 27 | 27 | 2021/05/05 20:09:00.000000000 | 20210505.0 | P0622 | 13641 | 1 | 5 | 4041596972700199 | 13.87 | 1 | 14.89 | 14.89 | 13.87 |
| 28 | 28 | 2021/05/06 05:08:00.000000000 | 20210506.0 | P0190 | 13238 | 6 | 3 | 5048374002788455 | 18.93 | 2 | 20.89 | 41.78 | 37.86 |
| 29 | 29 | 2021/05/06 05:16:00.000000000 | 20210506.0 | P0432 | 13462 | 6 | 7 | 5108756241400339 | 17.36 | 1 | 18.39 | 18.39 | 17.36 |
| 30 | 30 | 2021/05/06 05:58:00.000000000 | 20210506.0 | P0524 | 13549 | 3 | 4 | 374283854417458 | 10.66 | 2 | 11.59 | 23.18 | 21.32 |
| 31 | 31 | 2021/05/06 14:58:00.000000000 | 20210506.0 | P0042 | 13098 | 5 | 5 | 4041599935977144 | 19.59 | 1 | 21.39 | 21.39 | 19.59 |
| 32 | 32 | 2021/05/06 16:16:00.000000000 | 20210506.0 | P0474 | 13501 | 1 | 4 | 4041594587730164 | 15.66 | 2 | 17.49 | 34.98 | 31.32 |
| 33 | 33 | 2021/05/06 16:43:00.000000000 | 20210506.0 | P0500 | 13525 | 6 | 4 | 5048371164242156 | 7.53 | 1 | 8.29 | 8.29 | 7.53 |

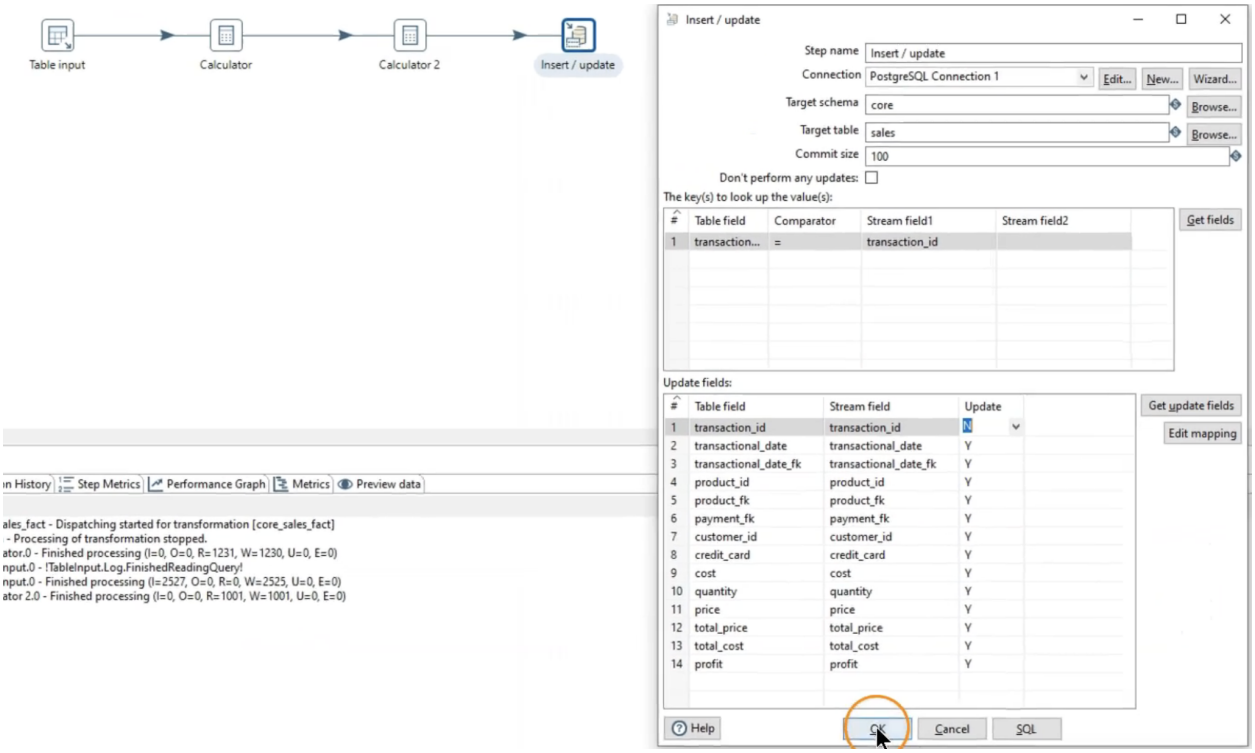
Close Stop Get more rows

- Set up another calculator to calculate profit



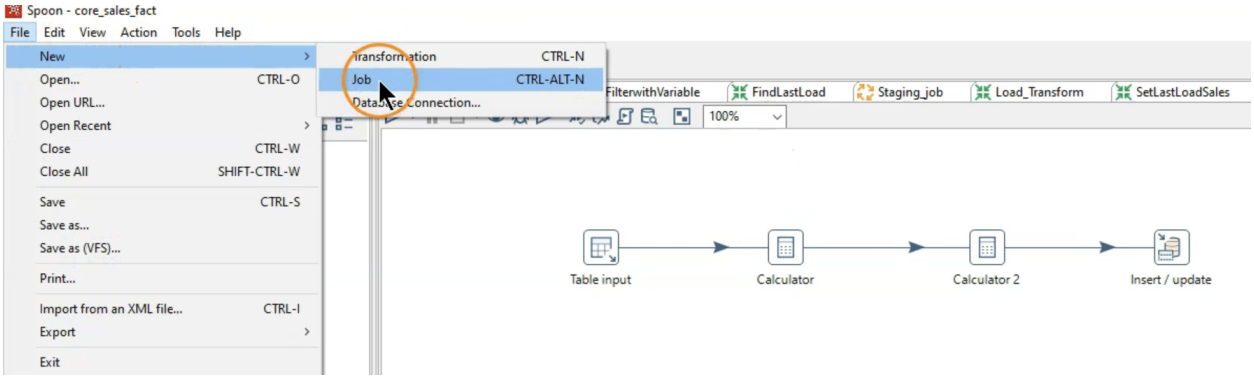


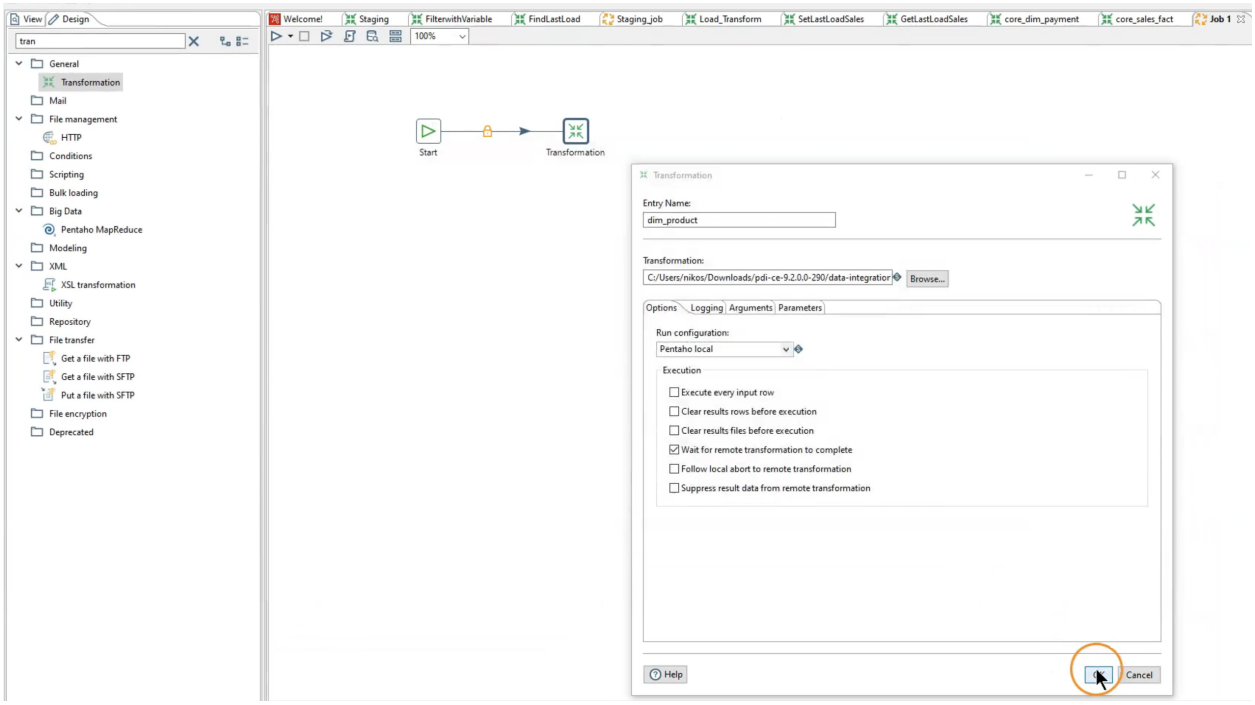

- Set up Insert/update. The transaction id should not be updated

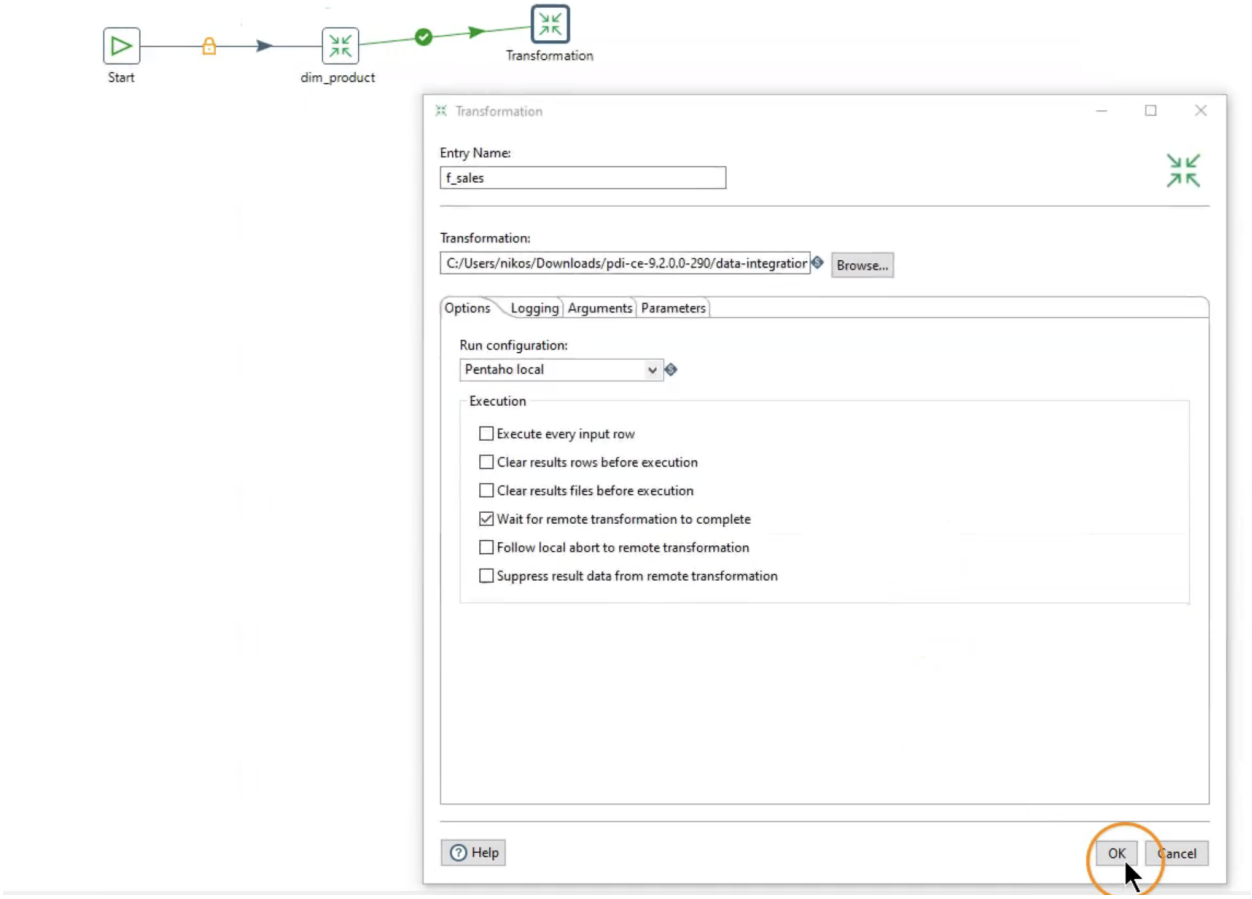


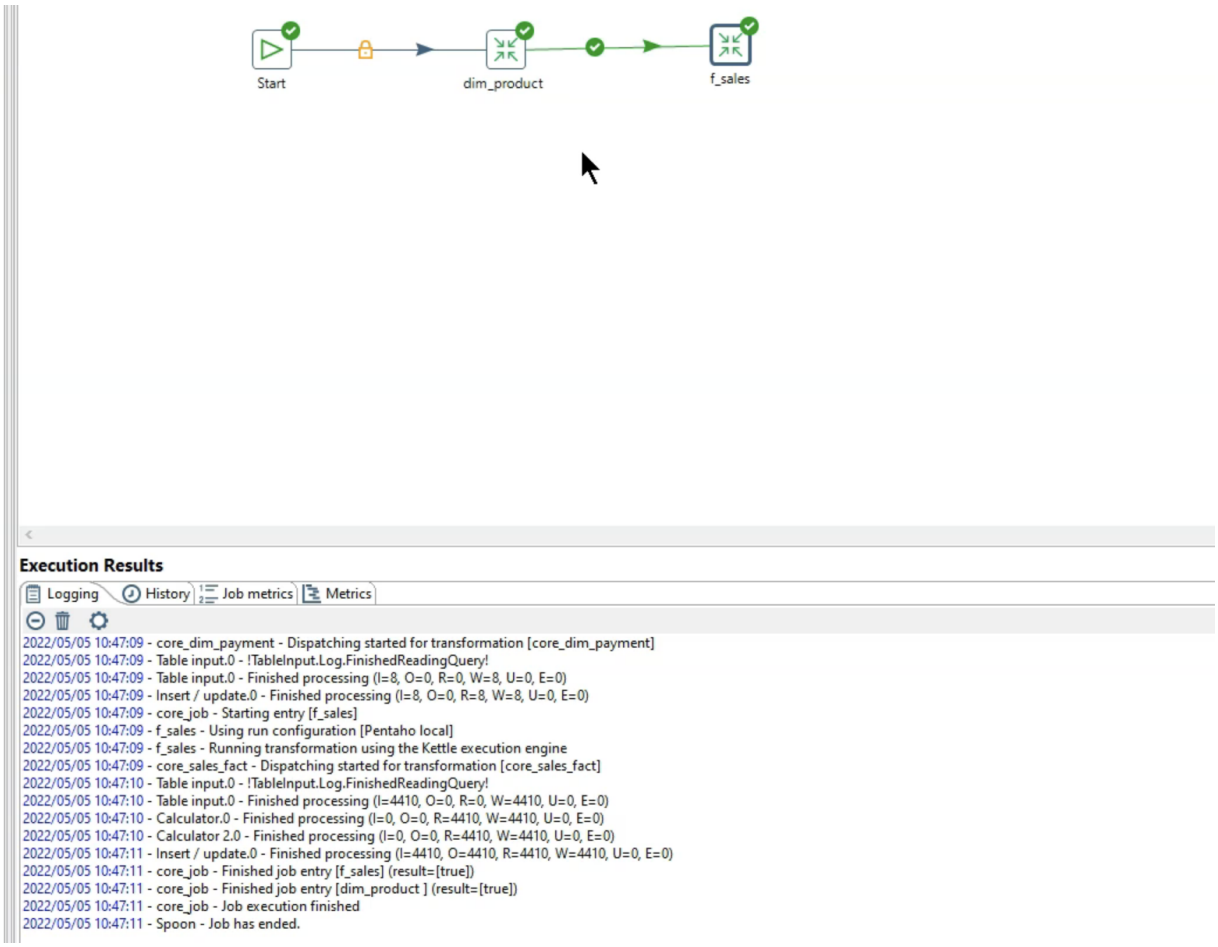

# Transform and load Job

- First let us create a new Job



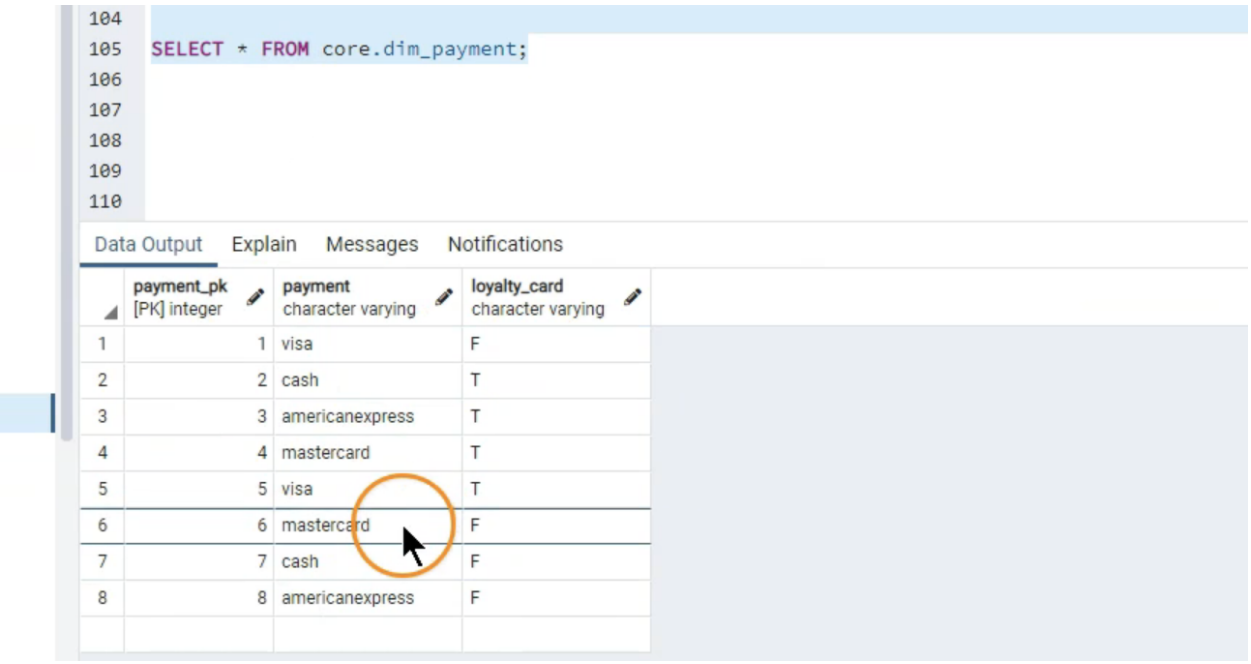

Start
dim_product
Transformation
Transformation
Entry Name:
f_sales
Transformation:
C:/Users/nikos/Downloads/pdi-ce-9.2.0.0-290/data-integration
Browse...
Options
Logging
Arguments
Parameters
Run configuration:
Pentaho local
Execution
Execute every input row
Clear results rows before execution
Clear results files before execution
Wait for remote transformation to complete
Follow local abort to remote transformation
Suppress result data from remote transformation
Help
OK
Cancel




```
102
103 SELECT * FROM core.sales;
104
105
```

Data Output | Explain | Messages | Notifications

| actional_date<br>timestamp without time zone | transactional_date_fk<br>bigint | product_id<br>character varying | product_fk<br>integer | customer_id<br>integer | payment_fk<br>integer | credit_card<br>bigint | cost<br>numeric | quantity<br>integer | price<br>numeric | total_cost<br>numeric | total_price<br>numeric | profit<br>numeric |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| -05-04 02:00:00 | 20210504 | P0494 | 13519 | 4 | 1 | 4041593010498829 | 17.33 | 2 | 18.29 | 34.66 | 36.58 | 1.92 |
| -05-04 03:04:00 | 20210504 | P0221 | 13267 | 5 | 1 | 4041596151234556 | 0.59 | 1 | 1.49 | 0.59 | 1.49 | 0.9 |
| -05-04 03:56:00 | 20210504 | P0625 | 13643 | 5 | 1 | 4041594885335898 | 5.15 | 3 | 5.89 | 15.45 | 17.67 | 2.22 |
| -05-04 05:20:00 | 20210504 | P0431 | 13461 | 8 | 6 | 5108753677552345 | 10.67 | 2 | 11.59 | 21.34 | 23.18 | 1.84 |
| -05-04 05:45:00 | 20210504 | P0058 | 13113 | 5 | 4 | 5108752372298261 | 11.38 | 2 | 12.39 | 22.76 | 24.78 | 2.02 |
| -05-04 06:58:00 | 20210504 | P0385 | 13416 | 6 | 8 | 374288563442549 | 13.22 | 1 | 14.69 | 13.22 | 14.69 | 1.47 |
| -05-04 07:03:00 | 20210504 | P0575 | 13596 | 4 | 1 | 4041598869758 | 2.81 | 1 | 3.99 | 2.81 | 3.99 | 1.18 |
| -05-04 07:45:00 | 20210504 | P0187 | 13235 | 5 | 8 | 374283491107967 | 4.17 | 1 | 4.89 | 4.17 | 4.89 | 0.72 |
| -05-04 09:58:00 | 20210504 | P0074 | 13129 | 7 | 6 | 5108753211196286 | 18.11 | 1 | 19.79 | 18.11 | 19.79 | 1.68 |
| -05-04 15:51:00 | 20210504 | P0456 | 13484 | 5 | 4 | 5048370523083285 | 18.35 | 2 | 20.19 | 36.7 | 40.38 | 3.68 |
| -05-04 16:13:00 | 20210504 | P0519 | 13544 | 5 | 6 | 5048379752512880 | 12.45 | 3 | 14.09 | 37.35 | 42.27 | 4.92 |
| -05-04 17:07:00 | 20210504 | P0071 | 13126 | 12 | 1 | 4041594984149257 | 13.09 | 5 | 14.69 | 65.45 | 73.45 | 8 |
| -05-04 19:57:00 | 20210504 | P0550 | 13574 | 5 | 8 | 374288299677079 | 15.35 | 1 | 17.09 | 15.35 | 17.09 | 1.74 |
| -05-04 23:06:00 | 20210504 | P0507 | 13532 | 4 | 6 | 5048376045855902 | 8.17 | 3 | 8.89 | 24.51 | 26.67 | 2.16 |



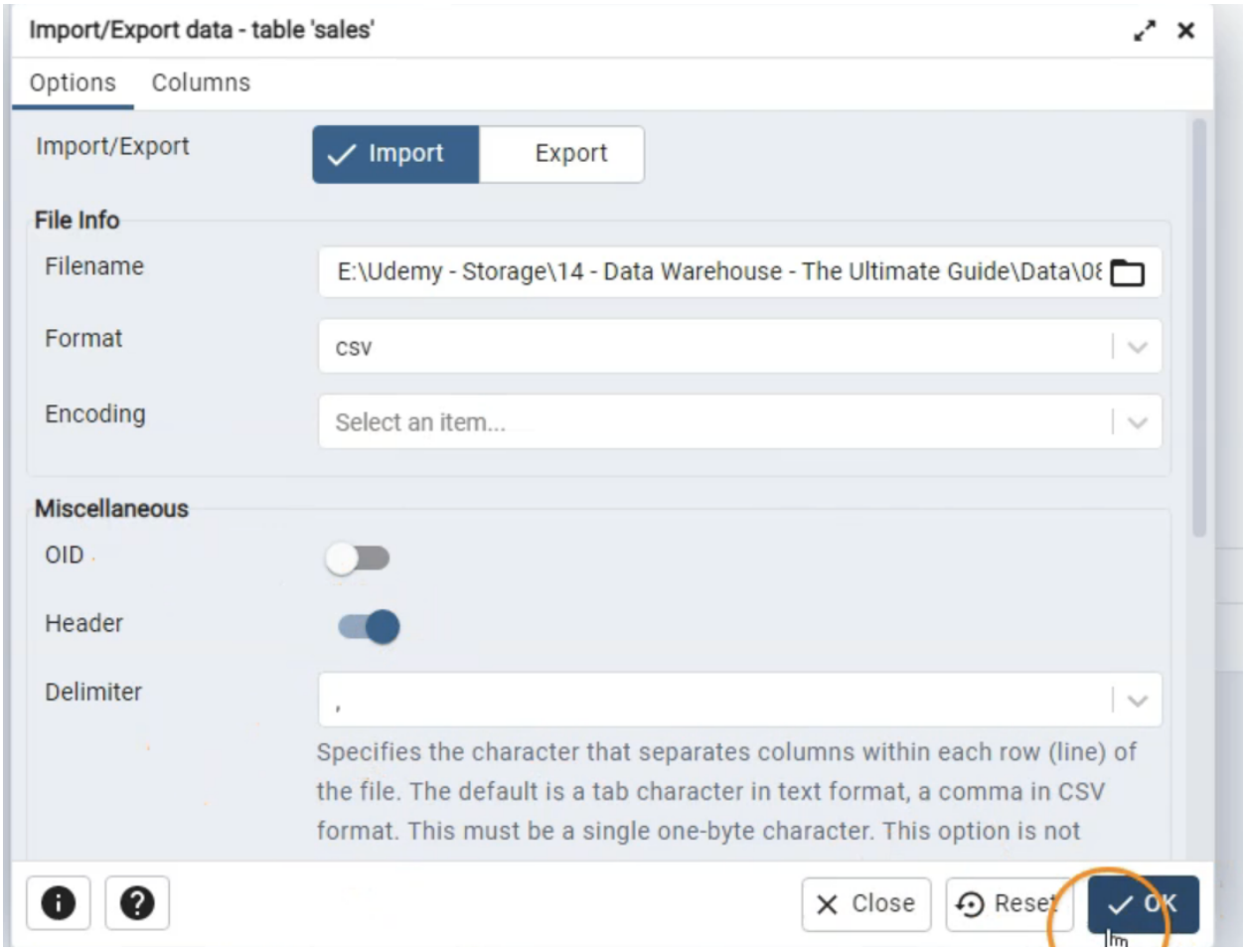
```
104
105 SELECT * FROM core.dim_payment;
106
107
108
109
110
```

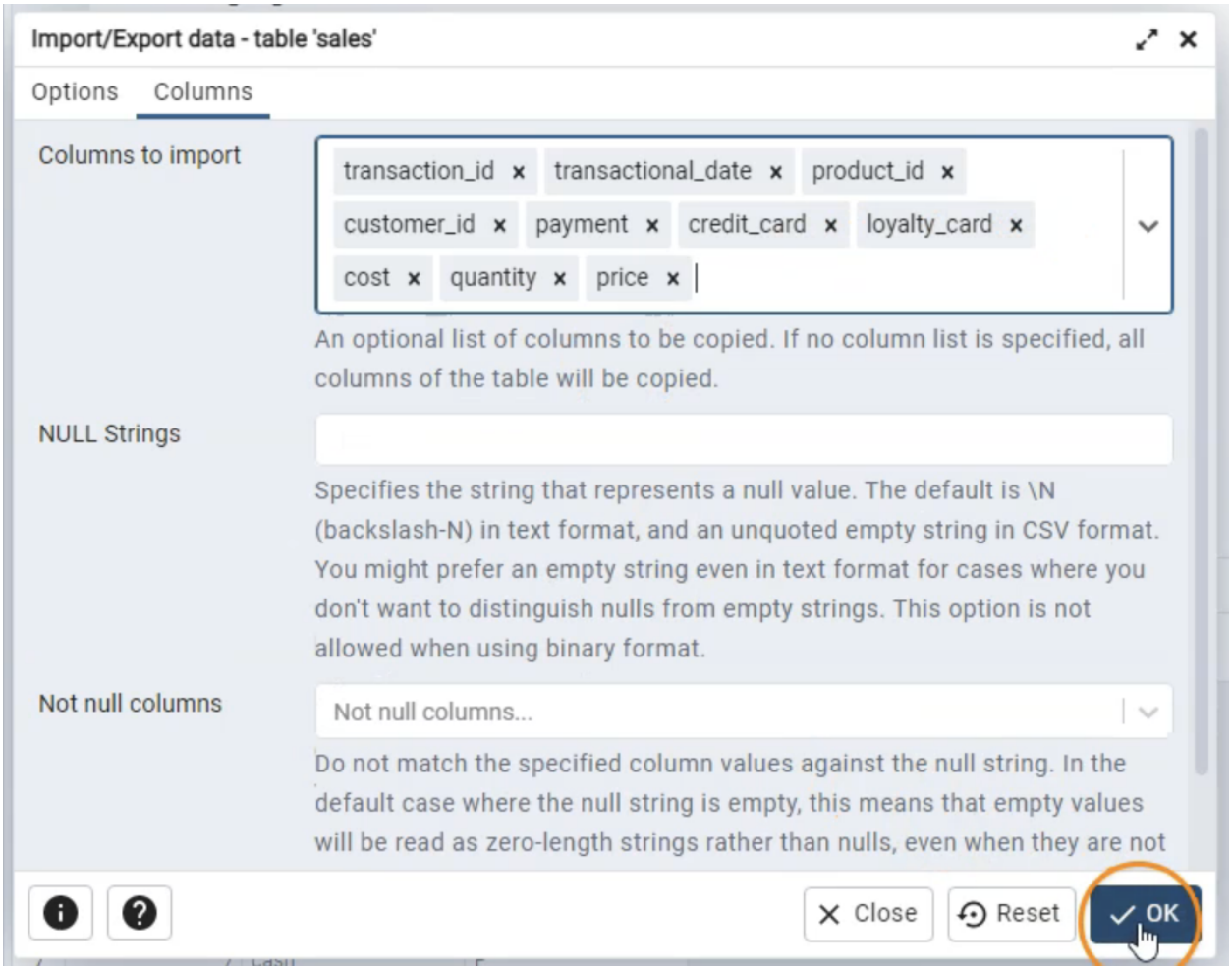
Data Output | Explain | Messages | Notifications

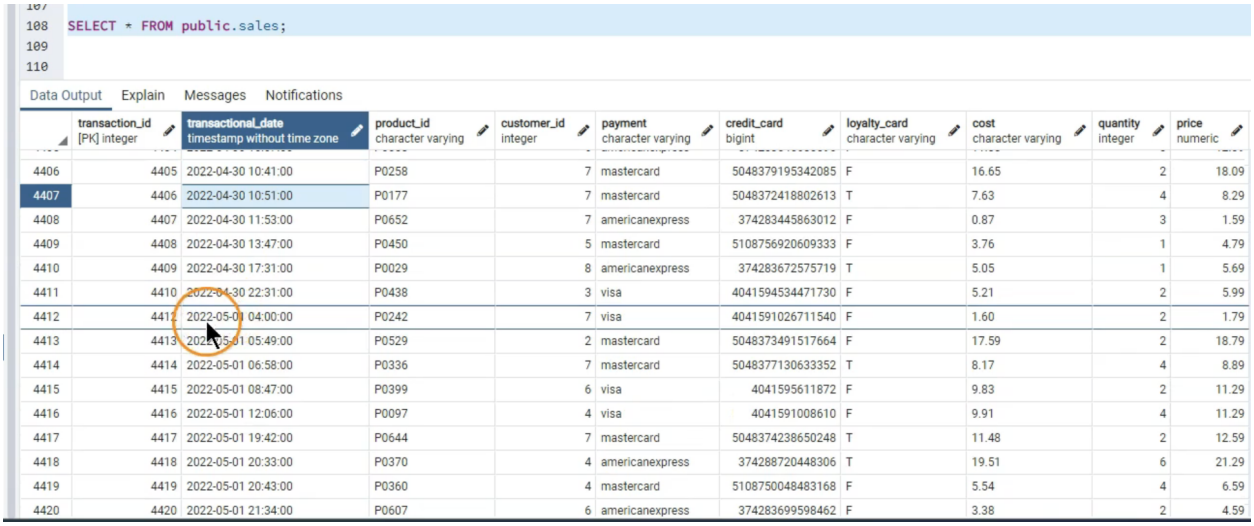
| | payment_pk [PK] integer | payment character varying | loyalty_card character varying |
|---|---|---|---|
| 1 | 1 | visa | F |
| 2 | 2 | cash | T |
| 3 | 3 | americanexpress | T |
| 4 | 4 | mastercard | T |
| 5 | 5 | visa | T |
| 6 | 6 | mastercard | F |
| 7 | 7 | cash | F |
| 8 | 8 | americanexpress | F |

# Final ETL Job and Incremental load

- Add some new data to our source table to check and set up how we can do delta load
- Also include all of the columns while importing the data as there is data already present

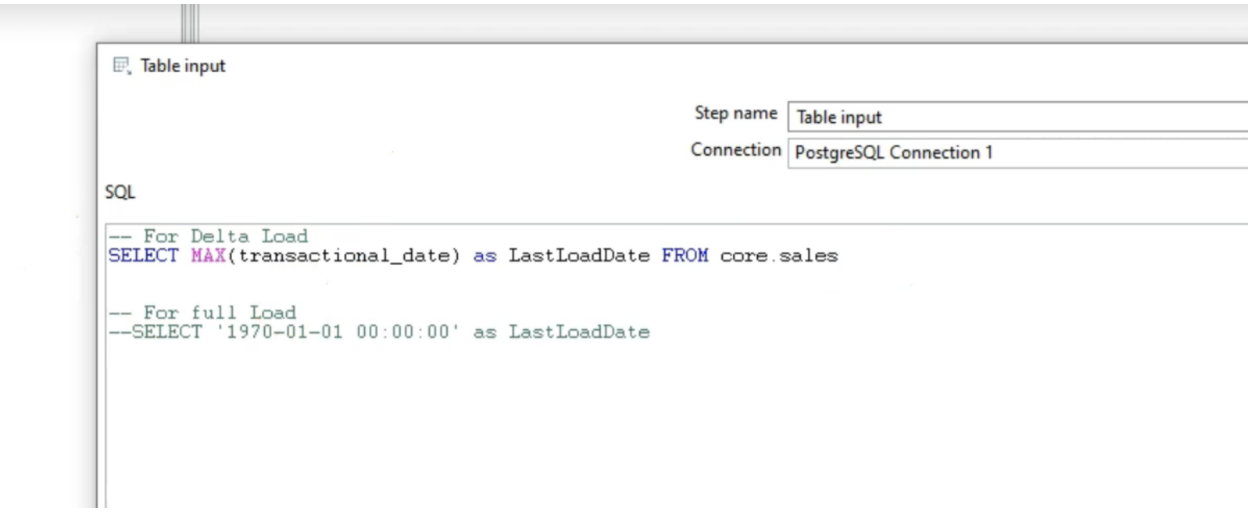

Import/Export data - table 'sales'
Options
Columns
Import/Export
Import
Export
File Info
Filename
E:\Udemy - Storage\14 - Data Warehouse - The Ultimate Guide\Data\0
Format
csv
Encoding
Select an item...
Miscellaneous
OID
Header
Delimiter
,
Specifies the character that separates columns within each row (line) of the file. The default is a tab character in text format, a comma in CSV format. This must be a single one-byte character. This option is not
Close
Reset
OK

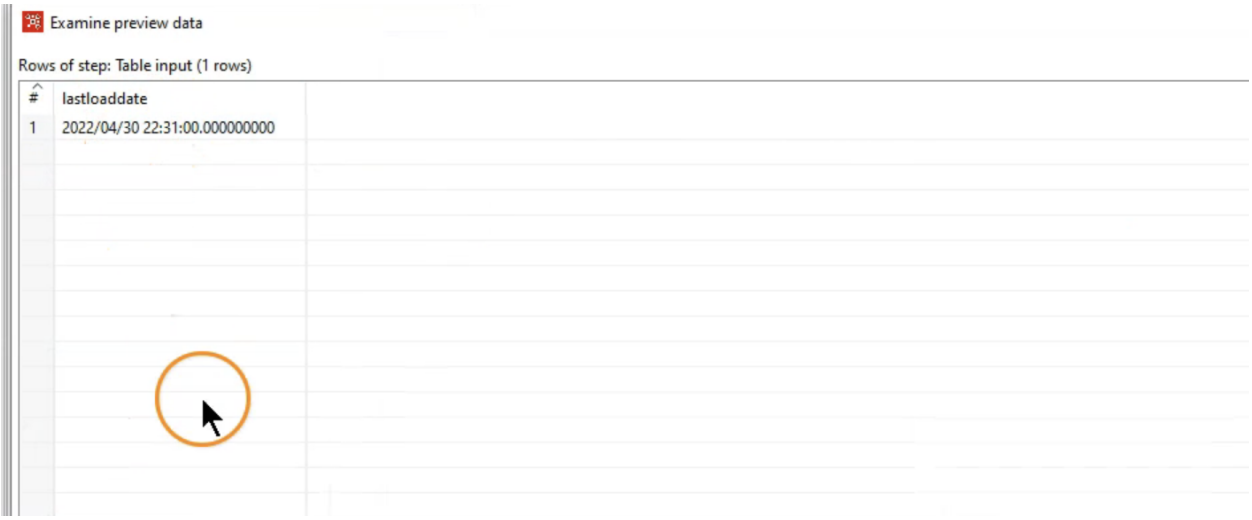


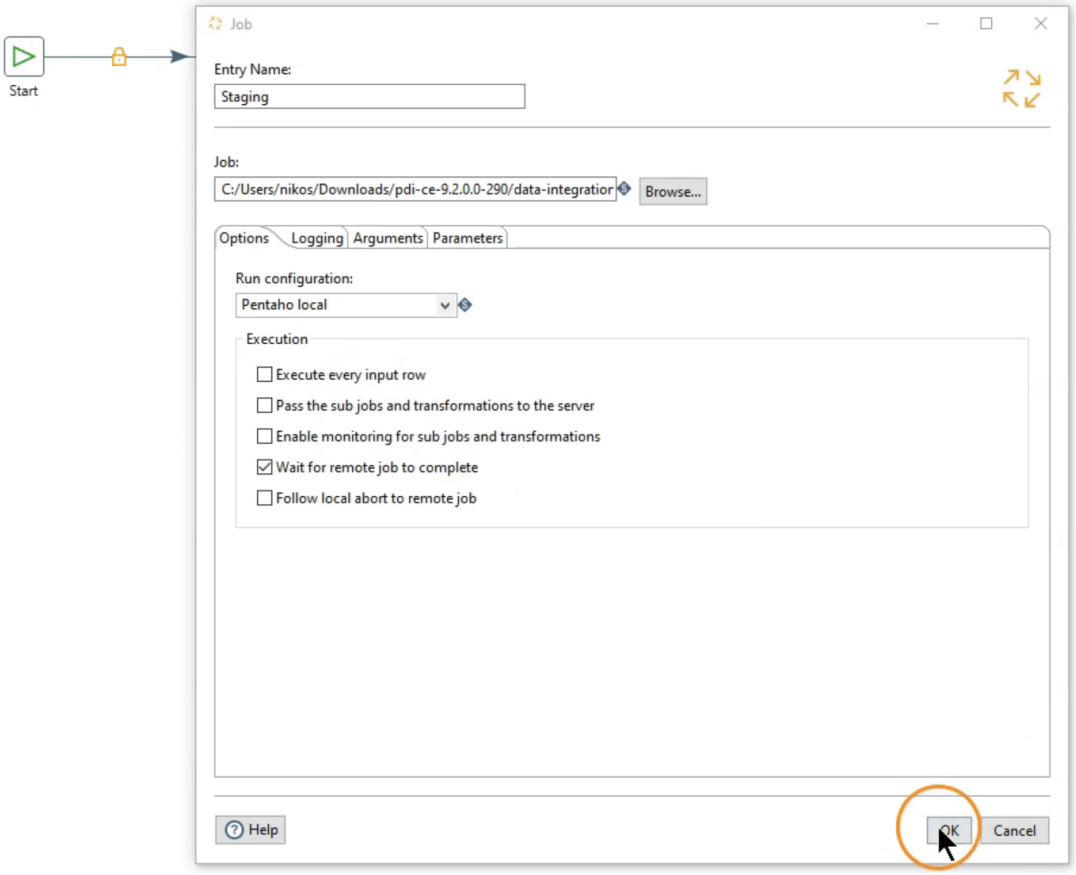

| | transaction_id [PK] integer | transactional_date timestamp without time zone | product_id character varying | customer_id integer | payment character varying | credit_card bigint | loyalty_card character varying | cost character varying | quantity integer | price numeric |
|---|---|---|---|---|---|---|---|---|---|---|
| 4406 | 4405 | 2022-04-30 10:41:00 | P0258 | 7 | mastercard | 5048379195342085 | F | 16.65 | 2 | 18.09 |
| 4407 | 4406 | 2022-04-30 10:51:00 | P0177 | 7 | mastercard | 5048372418802613 | T | 7.63 | 4 | 8.29 |
| 4408 | 4407 | 2022-04-30 11:53:00 | P0652 | 7 | americanexpress | 374283445863012 | F | 0.87 | 3 | 1.59 |
| 4409 | 4408 | 2022-04-30 13:47:00 | P0450 | 5 | mastercard | 5108756920609333 | F | 3.76 | 1 | 4.79 |
| 4410 | 4409 | 2022-04-30 17:31:00 | P0029 | 8 | americanexpress | 374283672575719 | T | 5.05 | 1 | 5.69 |
| 4411 | 4410 | 2022-04-30 22:31:00 | P0438 | 3 | visa | 4041594534471730 | F | 5.21 | 2 | 5.99 |
| 4412 | 4411 | 2022-05-01 04:00:00 | P0242 | 7 | visa | 4041591026711540 | F | 1.60 | 2 | 1.79 |
| 4413 | 4413 | 2022-05-01 05:49:00 | P0529 | 2 | mastercard | 5048373491517664 | F | 17.59 | 2 | 18.79 |
| 4414 | 4414 | 2022-05-01 06:58:00 | P0336 | 7 | mastercard | 5048377130633352 | T | 8.17 | 4 | 8.89 |
| 4415 | 4415 | 2022-05-01 08:47:00 | P0399 | 6 | visa | 4041595611872 | F | 9.83 | 2 | 11.29 |
| 4416 | 4416 | 2022-05-01 12:06:00 | P0097 | 4 | visa | 4041591008610 | F | 9.91 | 4 | 11.29 |
| 4417 | 4417 | 2022-05-01 19:42:00 | P0644 | 7 | mastercard | 5048374238650248 | T | 11.48 | 2 | 12.59 |
| 4418 | 4418 | 2022-05-01 20:33:00 | P0370 | 4 | americanexpress | 374288720448306 | T | 19.51 | 6 | 21.29 |
| 4419 | 4419 | 2022-05-01 20:43:00 | P0360 | 4 | mastercard | 5108750048483168 | F | 5.54 | 4 | 6.59 |
| 4420 | 4420 | 2022-05-01 21:34:00 | P0607 | 6 | americanexpress | 374283699598462 | F | 3.38 | 2 | 4.59 |

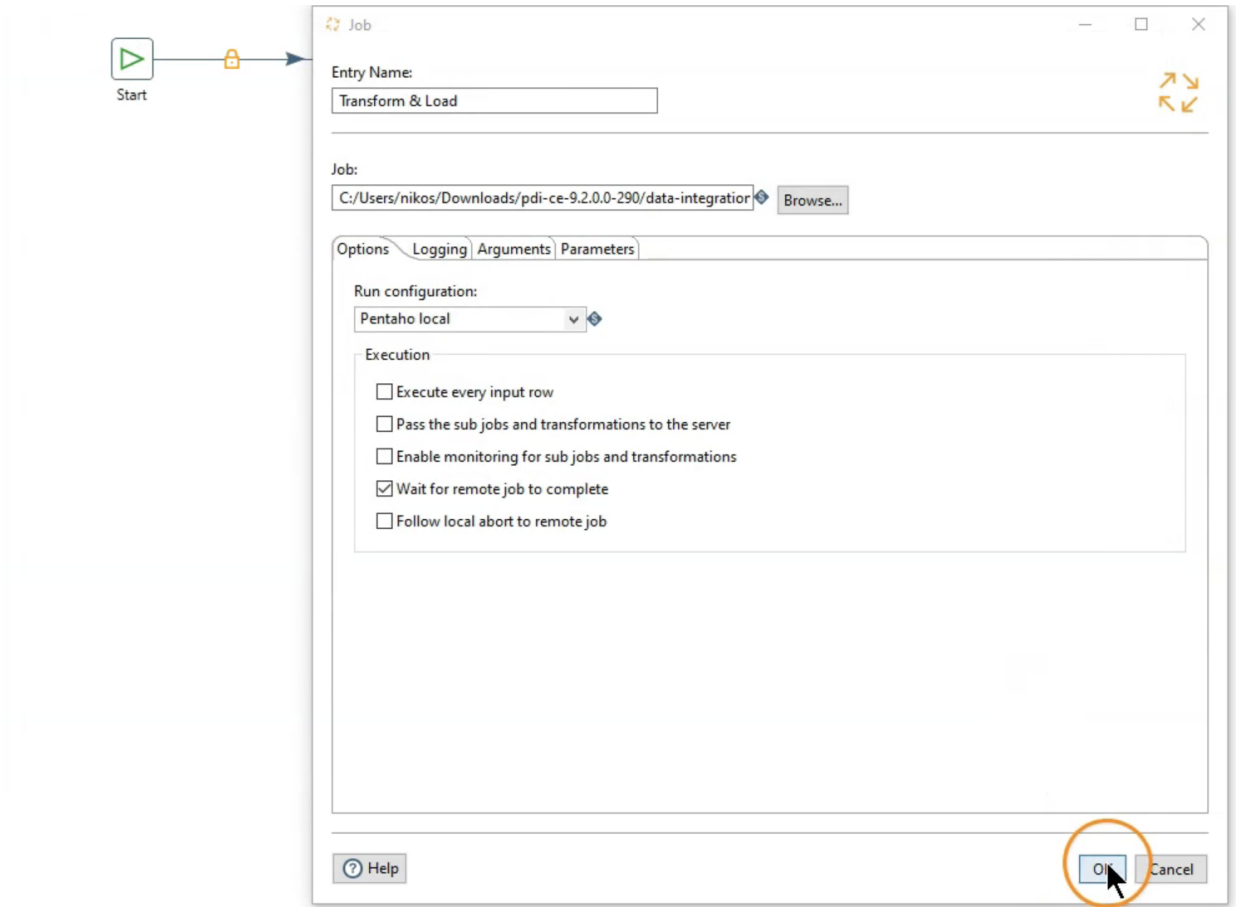
- Change the logic to obtain the max transaction date




- We can also set up a parent job which will call other 2 jobs in sequence


Start
Job
Entry Name:
Staging
Job:
C:/Users/nikos/Downloads/pdi-ce-9.2.0.0-290/data-integratior
Browse...
Options
Logging
Arguments
Parameters
Run configuration:
Pentaho local
Execution
Execute every input row
Pass the sub jobs and transformations to the server
Enable monitoring for sub jobs and transformations
Wait for remote job to complete
Follow local abort to remote job
Help
OK
Cancel


Start
Job
Entry Name:
Transform & Load
Job:
C:/Users/nikos/Downloads/pdi-ce-9.2.0.0-290/data-integratior
Browse...
Options
Logging
Arguments
Parameters
Run configuration:
Pentaho local
Execution
Execute every input row
Pass the sub jobs and transformations to the server
Enable monitoring for sub jobs and transformations
Wait for remote job to complete
Follow local abort to remote job
Help
OK
Cancel


**Execution Results**

Logging | History | Job metrics | Metrics

```
2022/05/05 10:54:49 - Table input.0 - Finished processing (I=6, O=0, R=0, W=6, U=0, E=0)
2022/05/05 10:54:49 - Insert / update.0 - Finished processing (I=6, O=0, R=6, W=6, U=0, E=0)
2022/05/05 10:54:49 - core_job - Starting entry [f_sales]
2022/05/05 10:54:49 - f_sales - Using run configuration [Pentaho local]
2022/05/05 10:54:49 - f_sales - Running transformation using the Kettle execution engine
2022/05/05 10:54:49 - core_sales_fact - Dispatching started for transformation [core_sales_fact]
2022/05/05 10:54:50 - Table input.0 - !TableInput.Log.FinishedReadingQuery!
2022/05/05 10:54:50 - Table input.0 - Finished processing (I=40, O=0, R=0, W=40, U=0, E=0)
2022/05/05 10:54:50 - Calculator.0 - Finished processing (I=0, O=0, R=40, W=40, U=0, E=0)
2022/05/05 10:54:50 - Calculator 2.0 - Finished processing (I=0, O=0, R=40, W=40, U=0, E=0)
2022/05/05 10:54:50 - Insert / update.0 - Finished processing (I=40, O=40, R=40, W=40, U=0, E=0)
2022/05/05 10:54:50 - core_job - Finished job entry [f_sales] (result=[true])
2022/05/05 10:54:50 - core_job - Finished job entry [dim_product ] (result=[true])
2022/05/05 10:54:50 - CompleteETLprocess - Finished job entry [Transform & Load] (result=[true])
2022/05/05 10:54:50 - CompleteETLprocess - Finished job entry [Staging] (result=[true])
2022/05/05 10:54:50 - CompleteETLprocess - Job execution finished
2022/05/05 10:54:50 - Spoon - Job has ended.
```

```
108 SELECT * FROM core.sales
109 order by transaction_id desc
110
```

Data Output | Explain | Messages | Notifications

| | transaction_id [PK] integer | transactional_date timestamp without time zone | transactional_date_fk bigint | product_id character varying | product_fk integer | customer_id integer | payment_fk integer | credit_card bigint | cost numer |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 4450 | 2022-05-03 23:20:00 | 20220503 | P0198 | 13245 | 9 | 6 | 5048375389506501 | |
| 2 | 4449 | 2022-05-03 22:46:00 | 20220503 | P0453 | 13481 | 7 | 1 | 4041594013905174 | |
| 3 | 4448 | 2022-05-03 21:31:00 | 20220503 | P0668 | 13683 | 3 | 8 | 374283719710394 | |
| 4 | 4447 | 2022-05-03 19:51:00 | 20220503 | P0690 | 13558 | 7 | 1 | 4041597439687276 | |
| 5 | 4446 | 2022-05-03 18:31:00 | 20220503 | P0524 | 13549 | 10 | 1 | 4041592104970 | |
| 6 | 4445 | 2022-05-03 18:09:00 | 20220503 | P0180 | 13228 | 7 | 8 | 374288919910694 | |
| 7 | 4444 | 2022-05-03 15:53:00 | 20220503 | P0272 | 13314 | 4 | 1 | 4041596413989 | |
| 8 | 4443 | 2022-05-03 14:43:00 | 20220503 | P0255 | 13298 | 7 | 1 | 4041591403548887 | |
| 9 | 4442 | 2022-05-03 11:07:00 | 20220503 | P0344 | 13378 | 4 | 4 | 5048372948984931 | |
| 10 | 4441 | 2022-05-03 10:37:00 | 20220503 | P0066 | 13121 | 7 | 1 | 4041593485881 | |