

# Testy Funkcjonalne

Zespół Danonkowych Żółwi Ninja 🐢:

Katarzyna Solawa i Mateusz Sperkowski

## Generowanie Danych

Poniżej widzimy część kodu wykorzystanego do generowania częściowo losowo danych oraz tego skutki.

```
Data_generation.ipynb
[3]: sum_days_since_prior = orders[['user_id', 'days_since_prior_order']].groupby(['user_id']).sum()
    day_of_last_order = date(2017, 1, 1) - timedelta(days=1.0)
    first_day = date(2016, 1, 1)
    sum_days_since_prior['first_order_date'] = sum_days_since_prior['days_since_prior_order'].apply(lambda x: day_of_last_order - timedelta(days=x))
    sum_days_since_prior['first_order_date'] = sum_days_since_prior['first_order_date'].apply(lambda x: random_date(first_day, x))
    sum_days_since_prior = sum_days_since_prior.reset_index()
    ddd = WeatherEvents[['City']].unique()
    cities_singular = uscities['city'].value_counts()[uscities['city'].value_counts() == 1].index
    overlapping_datasets_cities=ddd[np.isin(ddd, cities_singular)]
    sum_days_since_prior['user_location'] = np.random.choice(overlapping_datasets_cities, size=len(sum_days_since_prior))
    orders['days_since_first_order']=orders.groupby(['user_id'])['days_since_prior_order'].cumsum()

[4]: sum_days_since_prior

[4]:
```

	user_id	days_since_prior_order	first_order_date	user_location
0	1	190.0	2016-04-23	Angleton
1	2	228.0	2016-03-10	Erlanger
2	3	144.0	2016-05-09	West Palm Beach
3	4	85.0	2016-01-07	Rancho Cordova
4	5	46.0	2016-02-21	Cloquet
...	...	...	...	...
206204	206205	50.0	2016-08-29	Kaycee
206205	206206	249.0	2016-02-07	Fort Stockton
206206	206207	229.0	2016-01-04	North Clarendon
206207	206208	361.0	2016-01-04	New Roads
206208	206209	240.0	2016-03-11	Jordan Valley

```

[5]: orders = pd.merge(orders, sum_days_since_prior[['user_id', 'first_order_date', 'user_location']], on='user_id')
    orders

[5]:
```

	order_id	user_id	eval_set	order_number	order_dow	order_hour_of_day	days_since_prior_order	days_since_first_order	first_order_date	user_location
0	2539329	1	prior	1	2	8	0.0	0.0	2016-04-23	Angleton
1	2398795	1	prior	2	3	7	15.0	15.0	2016-04-23	Angleton
2	473747	1	prior	3	3	12	21.0	36.0	2016-04-23	Angleton
3	2254736	1	prior	4	4	7	29.0	65.0	2016-04-23	Angleton
4	431534	1	prior	5	4	15	28.0	93.0	2016-04-23	Angleton
...	...	...	...	...	...	...	...	...	...	...
3421078	2266710	206209	prior	10	5	18	29.0	155.0	2016-03-11	Jordan Valley
3421079	1854736	206209	prior	11	4	10	30.0	185.0	2016-03-11	Jordan Valley
3421080	626363	206209	prior	12	1	12	18.0	203.0	2016-03-11	Jordan Valley
3421081	2977660	206209	prior	13	1	12	7.0	210.0	2016-03-11	Jordan Valley
3421082	272231	206209	train	14	6	14	30.0	240.0	2016-03-11	Jordan Valley

3421083 rows x 10 columns

```

[6]: cities_singular = uscities['city'].value_counts()[uscities['city'].value_counts() == 1].index

[7]: orders['order_date'] = orders.apply(lambda x: x['first_order_date'] + timedelta(days=x['days_since_first_order']), axis=1)

[9]: orders.to_csv('flat_sources/orders2.csv')

[11]: WeatherEvents = nd.read_csv("flat_sources/WeatherEvents Jan2016-Dec2021.csv")
```

```
Data_generation.ipynb X users_generation.ipynb Python 3

[105]: import pandas as pd

[113]: orders = pd.read_csv("orders.csv")

[114]: orders["user_id"].unique()

[114]: array([ 1, 2, 3, ..., 206207, 206208, 206209], dtype=int64)

[116]: len(orders["user_id"].unique())

[116]: 206209

[ ]: !pip install randomuser

[ ]: from randomuser import RandomUser
import time

[ ]: user_list = []

[ ]: while(len(user_list)< 206209):
time.sleep(60)
user_list +=RandomUser.generate_users(5000)
print(len(user_list))

[ ]: len(user_list)

[ ]: user_list = user_list[0:206209]

[ ]: data = pd.DataFrame({"User": user_list})

Data_generation.ipynb X users_generation.ipynb Python 3

[ ]: data =data[["first_name","last_name","gender","email","phone","age"]]

[ ]: data['User_id'] = range(1, len(data) + 1)

[ ]: data.to_csv("users.csv",index=False)

[111]: df = pd.read_csv("users.csv")

[112]: df

[112]:
```

	first_name	last_name	gender	email	phone	age	User_id
0	Dioclene	da Costa	female	dioclene.dacosta@example.com	(43) 1818-7195	47.0	1
1	Lucile	Martin	female	lucile.martin@example.com	05-96-50-82-61	34.0	2
2	Lisandro	Melo	male	lisandro.melo@example.com	(97) 8182-5106	29.0	3
3	John	Calvo	male	john.calvo@example.com	968-243-221	55.0	4
4	Angel	Leon	male	angel.leon@example.com	985-906-622	47.0	5
...	...	...	...	...	...	...	...
206204	Hanna	Sullivan	female	hanna.sullivan@example.com	041-533-6077	42.0	206205
206205	Lenni	Lehtinen	male	lenni.lehtinen@example.com	06-078-751	36.0	206206
206206	Hilla	Raisanen	female	hilla.raisanen@example.com	07-612-527	44.0	206207
206207	Björn	Gravem	male	bjornar.gravem@example.com	73453068	34.0	206208
206208	Oline	Bohler	female	oline.bohler@example.com	33036052	33.0	206209

206209 rows x 7 columns

## ETL

Poniżej widzimy proces etl w oprogramowaniu Tableau Prep, z zbliżeniami na niektóre z ciekawszych transformacji.

Tableau Prep Builder - ETL\_flow

File Edit Flow Server Help

Alerts (0)

Tableau Prep Builder - ETL\_flow

File Edit Flow Server Help

Alerts (0)

90%

Clean cities 11 fields 30K rows

Filter Values... Rename Fields... Create Calculated Field... 4 Recommendations Search

Changes (17)

- Rename Field [state\_abbreviation] From [state\_id] to [state\_abbreviation]
- Rename Field [county\_fips\_codes] From [county\_fips] to [county\_fips\_codes]
- Rename Field [latitude] From [lat] to [latitude]
- Rename Field [longitude] From [lng] to [longitude]
- Rename Field [latitude] From [latitude] to [latitude]
- Remove Field [source]
- Remove Field [incorporated]
- Remove Field [timezone]
- Remove Field [ranking]
- Rename Field [city\_id] From [id] to [city\_id]

is_military_city	city_name	state_abbreviation	state_name	county_fips_codes	county_name	latitude	longitude	population	density	city_id
False	New York	NY	New York	36,081	Queens	40.6943	-73.9249	18,680,025	10,768	1,840,034,016
False	Los Angeles	CA	California	6,037	Los Angeles	34.1141	-118.4068	12,531,334	3,267	1,840,020,491
False	Chicago	IL	Illinois	17,031	Cook	41.8375	-87.6866	8,586,888	4,576	1,840,000,494
False	Miami	FL	Florida	12,086	Miami-Dade	25.784	-80.2101	6,076,316	4,945	1,840,015,149
False	Dallas	TX	Texas	48,113	Dallas	32.7935	-96.7667	5,910,669	1,522	1,840,019,440

Tableau Prep Builder - ETL\_flow

File Edit Flow Server Help

Alerts (0)

90%

Products Table 4 fields

Save output to Database table

Settings Custom SQL

Connection DESKTOP-2F87DE8

Database DWH\_Project

Table Products

Write Options

Full refresh Replace data

Incremental refresh Append to table

Run Flow

Write 4 of 6 fields to the table "Products" in DESKTOP-2F87DE8

Preview Full refresh

aisle	product_id	product_name	department
cookies cakes	1	Chocolate Sandwich Cookies	snacks
spices seasonings	2	All-Seasons Salt	pantry
tea	3	Robust Golden Unsweetened Oolong Tea	beverages
frozen meals	4	Smart Ones Classic Favorites Mini Rigato	frozen
marinades meat preparation	5	Green Chile Anytime Sauce	pantry
cold flu allergy	6	Dry Nose Oil	personal care
juice nectars	7	Pure Coconut Water With Orange	beverages
frozen produce	8	Cut Russet Potatoes Steam N' Mash	frozen
yogurt	9	Light Strawberry Blueberry Yogurt	dairy eggs
water seltzer sparkling water	10	Sparkling Orange Juice & Prickly Pear Be	beverages
refrigerated	11	Peach Mango Juice	beverages
frozen dessert	12	Chocolate Fudge Layer Cake	frozen
cold flu allergy	13	Saline Nasal Mist	personal care
dish detergents	14	Fresh Scent Dishwasher Cleaner	household
diapers wipes	15	Overnight Diapers Size 6	babies
ice cream toppings	16	Mint Chocolate Flavored Syrup	snacks
poultry counter	17	Rendered Duck Fat	meat seafood
frozen pizza	18	Pizza for One Suprema Frozen Pizza	frozen

Tableau Prep Builder - ETL\_flow

File Edit Flow Server Help

Alerts (0)

90%

Input

Settings Multiple Files Data Sample Changes (4)

Character Set UTF-8

Locale English (United States)

Incremental Refresh

Specify the input and output fields to use to identify new rows.

☒ Enable incremental refresh

Input field # User\_id

Output Users Table

Select the output and field with the last processed value for field "User\_id".

Output field # user\_id

users 7 fields | Filter Values...

Clear the checkbox to remove fields. You can also filter your data or change data types. [Add a clean step](#) to view and clean data.

Fields selected: 3 of 7

	Type	Field Name	Original Field Name	Changes	Preview
<input type="checkbox"/>	Abc	first_name	first_name		Dioclene, Lucile, Lisandro
<input type="checkbox"/>	Abc	last_name	last_name		da Costa, Martin, Melo
<input checked="" type="checkbox"/>	Abc	gender	gender		female, male
<input type="checkbox"/>	Abc	email	email		dioclene.dacosta@example.com, lucile.martin@example.com, lisand...
<input type="checkbox"/>	Abc	phone	phone		(43) 1818-7195, 05-96-50-82-61, (97) 8182-5106
<input checked="" type="checkbox"/>	#	age	age		47, 34, 29
<input checked="" type="checkbox"/>	#	User_id	User_id		1, 2, 3

Tableau Prep Builder - ETL\_flow

File Edit Flow Server Help

order\_product... full\_order\_pro... Remove Dups Cities WeatherEvent... Make data\_ids

full\_order\_products 4 fields 1M rows Filter Values... Create Calculated Field...

Settings Changes (3)

- Union 2 inputs
- Remove Field [Table Names]
- Calculated Field [is\_reordered] IF [reordered] = 1 THEN 'TRUE' ELSE 'FALSE' END
- Remove Field [reordered]

Union Results Show only mismatched fields

is_reordered	order_id	product_id	add_to_cart_order
TRUE	6,985	38,200	9
TRUE	6,985	23,909	10
TRUE	6,985	37,766	11
TRUE	6,985	9,175	12
TRUE	6,985	29,095	13
TRUE	6,985	22,959	14
TRUE	6,985	29,487	15
TRUE	6,985	1,661	16
TRUE	6,985	43,112	17
TRUE	6,985	19,246	18
TRUE	6,985	16,965	19
TRUE	6,985	47,890	20
TRUE	6,985	40,232	21
TRUE	6,985	8,424	22
TRUE	6,985	34,533	23
TRUE	6,985	35,825	24
TRUE	6,985	6,178	25
TRUE	6,985	36,393	26

Tableau Prep Builder - ETL\_flow

File Edit Flow Server Help

Orders\_transf... Date\_to\_key Orders\_facts Cities foreign k... Make Primary... Left join Clean nulls Output

Orders\_facts 11 fields 530K rows Filter Values... Create Calculated Field...

Settings Changes (0)

- Join [order\_id] == [order\_id]

Join Clauses Show only mismatched values

Date_to_key	Remove Dups
order_id	order_id
6	1
14	2
15	3
17	4
22	5
34	6
37	7
38	8
42	9
43	10
45	11
46	12
52	13
56	14
57	15
60	16
61	18
64	19

Join Results

is_reordered	order_date_id	order_id	user_id
TRUE	20,161,126	3,797	6,577
TRUE	20,161,216	78,546	23,357
TRUE	20,160,124	51,023	22,560
TRUE	20,160,216	95,781	65,558
TRUE	20,160,625	12,406	32,270
FALSE	20,160,807	34,861	29,840
TRUE	20,160,420	543,228	35,797
TRUE	20,160,513	36,841	48,194
TRUE	20,160,322	38,968	40,043
TRUE	20,160,922	101,868	94,572
TRUE	20,161,224	35,770	35,572
TRUE	20,161,017	758,378	4,235
TRUE	20,160,527	14,159	38,291
TRUE	20,161,022	99,385	31,770
FALSE	20,160,405	19,297	25,261
TRUE	20,160,508	59,408	16,978
TRUE	20,161,227	2,159,350	23,230
TRUE	20,160,612	1,066	16,408

Tableau Prep Builder - ETL\_flow

File Edit Flow Server Help

Alerts (0)

90%

Cities foreign key 11 fields 567K rows

Settings

- Remove Field [order\_id-1]
- Join [user\_location] == [city]
- Remove Field [city]

Changes (2)

Join Clauses Show only mismatched values

Orders\_facts

- ! user\_location

Cities

- Aberdeen Proving Ground
- Accident
- Aiken
- Alabaster
- Alamogordo
- Alamosa
- Albert Lea
- Albin
- Albuquerque
- Alcoa
- Alexander City
- Amargosa Valley
- Amarillo
- Angel Fire
- Angleton
- Ankeny
- Ann Arbor

Join Results

is_reordered	order_date_id	order_id	user_id
FALSE	20,161,009	100,618	48,873
TRUE	20,161,212	1,218,763	45,400
FALSE	20,160,807	32,729	14,783
TRUE	20,160,902	97,458	44,896
TRUE	20,160,402	87,706	23,886
TRUE	20,160,712	36,597	41,530
TRUE	20,160,408	4,130	44,229
TRUE	20,160,612	96,003	12,157
FALSE	20,160,123	63,228	82,749
TRUE	20,160,630	24,922	63,734
TRUE	20,160,802	535,076	50,504
TRUE	20,160,610	11,134	22,606
TRUE	20,160,213	32,513	23,087
TRUE	20,160,305	57,807	48,315
TRUE	20,161,230	2,469,410	7,342
TRUE	20,160,416	1,003,556	4,233

Tableau Prep Builder - ETL\_flow

File Edit Flow Server Help

Alerts (0)

90%

Clean nulls 15 fields 568K rows

Remove Field [user\_location]

Changes (4)

- Group Values [weather\_type] null replaced by "Clear"
- Group Values [severity] 3 values replaced by "Not Applicable"
- Group Values [precipitation\_inches] null replaced by "0"
- Remove Field [user\_location]

is_reordered	order_product_id	order_date_id	order_id	user_id	order_number	order_hour_of...
TRUE	10,887,327,548	20,160,612	108,873	33,380	16	8
FALSE	10,851,713,819	20,160,102	108,517	53,030	1	9
TRUE	255,588,836,436	20,161,124	2,555,888	30,388	11	17
TRUE	256,150,905,025	20,160,908	2,561,509	46,260	17	16
TRUE	6,841,226,209	20,160,407	68,412	36,593	9	10
TRUE	230,730,726,047	20,161,015	2,307,307	24,317	6	15
TRUE	164,633,639,275	20,161,224	1,646,336	36,110	30	18
TRUE	8,489,720,842	20,161,225	84,897	66,556	53	8
TRUE	10,937,511,984	20,161,210	109,375	15,060	95	13
TRUE	22,154,616,083	20,161,005	221,546	45,058	22	16
FALSE	10,935,509,755	20,160,505	109,355	78,341	3	17
FALSE	162,395,245,007	20,160,908	1,623,952	25,266	15	13
TRUE	6,333,534,358	20,160,704	63,335	22,909	23	17
TRUE	247,512,821,903	20,161,230	2,475,128	40,981	22	10
TRUE	174,546,443,772	20,160,819	1,745,464	56,254	5	15
TRUE	77,208,804,957	20,160,819	772,088	36,364	4	14
TRUE	8,680,404,261	20,160,519	86,804	14,740	7	17
FALSE	6,863,125,199	20,160,310	68,631	58,666	2	18

Tableau Prep Builder - ETL\_flow

File Edit Flow Server Help

Alerts (0)

90%

Input

Settings Multiple Files Data Sample Changes (6)

UTF-8

Locale English (United States)

**Incremental Refresh**  
Specify the input and output fields to use to identify new rows.

☒ Enable incremental refresh

Input field  
# order\_id

Output  
Output

Select the output and field with the last processed value for field "order\_id".

Output field  
# order\_id

**Orders\_transformed** 12 fields Filter Values...

Clear the check box to remove fields. You can also filter your data or change data types. [Add a clean step](#) to view and clean data.

Fields selected: 6 of 12

<input type="checkbox"/>	Type	Field Name	Original Field Name	Changes	Preview
<input type="checkbox"/>	#	F1	F1	<input type="checkbox"/>	0, 1, 2
<input checked="" type="checkbox"/>	#	order_id	order_id		2,539,329, 2,398,795, 473,747
<input checked="" type="checkbox"/>	#	user_id	user_id		1
<input type="checkbox"/>	Abc	eval_set	eval_set	<input type="checkbox"/>	prior
<input checked="" type="checkbox"/>	#	order_number	order_number		1, 2, 3
<input type="checkbox"/>	#	order_dow	order_dow	<input type="checkbox"/>	2, 3
<input checked="" type="checkbox"/>	#	order_hour_of...	order_hour_of_day		8, 7, 12
<input type="checkbox"/>	#	days_since_prio...	days_since_prior_order	<input type="checkbox"/>	0, 15, 21
<input type="checkbox"/>	#	days_since_firs...	days_since_first_order	<input type="checkbox"/>	0, 15, 36
<input type="checkbox"/>	Abc	first_order_date	first_order_date	<input type="checkbox"/>	04/23/2016
<input checked="" type="checkbox"/>	Abc	user_location	user_location		Angleton
<input checked="" type="checkbox"/>	Abc	order_date	order_date		04/23/2016, 05/08/2016, 05/29/2016

## Hurtownia Danych

Poniżej widzimy sktypty w języku SQL wykorzystane do stworzenia hurtowni danych.

Microsoft SQL Server Management Studio interface showing two queries being executed on a SQL Server instance.

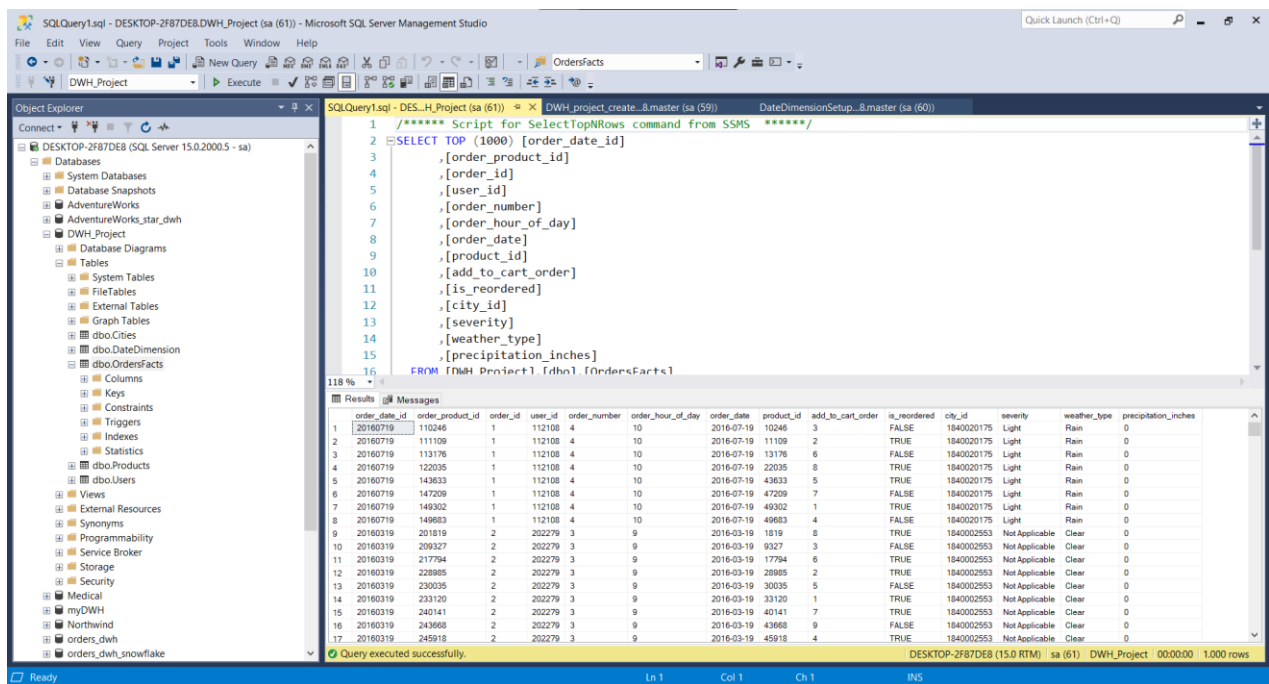
**Query 1: DateDimensionSetup.sql**

```
1 --https://www.mssqltips.com/sqlservertip/4054/creating-a-date-dimension-or-calendar-table-in-sql-server/
2 USE DWH_Project
3 GO
4
5 DECLARE @StartDate DATE = '19000101', @NumberOfYears INT = 200;
6
7 -- prevent set or regional settings from interfering with
8 -- interpretation of dates / literals
9
10 SET DATEFIRST 7;
11 SET DATEFORMAT mdy;
12 SET LANGUAGE US_ENGLISH;
13
14 DECLARE @CutoffDate DATE = DATEADD(YEAR, @NumberOfYears, @StartDate);
15
16 -- this is just a holding table for intermediate calculations:
17
18 CREATE TABLE #dim
19 (
20     [date] DATE PRIMARY KEY,
21     [day] AS DATEPART(DAY, [date]),
22     [month] AS DATEPART(MONTH, [date]),
23     FirstOfMonth AS CONVERT(DATE, DATEADD(MONTH, DATEDIFF(MONTH, 0, [date]), 0)),
24     [MonthName] AS DATENAME(MONTH, [date]),
25     [week] AS DATEPART(WEEK, [date]),
26     [ISOweek] AS DATEPART(ISO_WEEK, [date]),
27     [DayOfWeek] AS DATEPART(WEEKDAY, [date]),
28     [quarter] AS DATEPART(QUARTER, [date]),
29     [year] AS DATEPART(YEAR, [date]),
30     FirstOfYear AS CONVERT(DATE, DATEADD(YEAR, DATEDIFF(YEAR, 0, [date]), 0)),
31     Style112 AS CONVERT(CHAR(8), [date], 112),
32 )
```

**Query 2: DWH\_project\_create.sql**

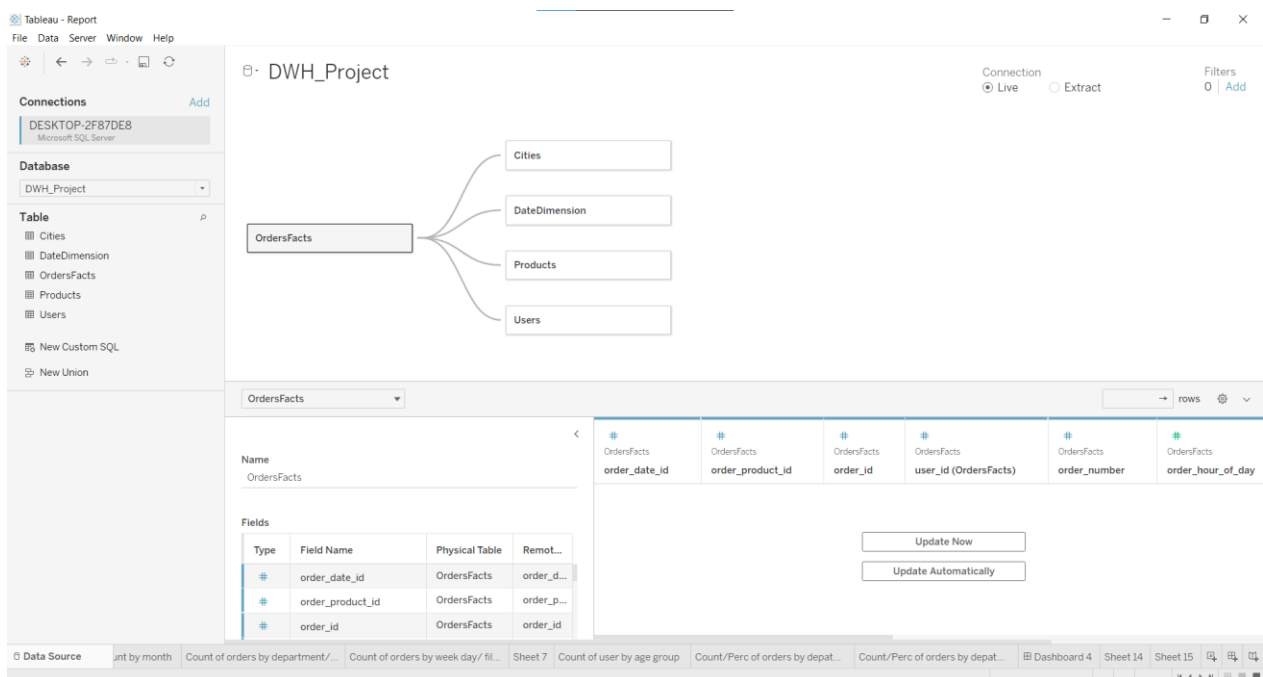
```
1 USE [DWH_Project]
2 GO
3
4 /***** Object: Table [dbo].[Cities] Script Date: 12.06.2022 13:18:11 *****/
5 SET ANSI_NULLS ON
6 GO
7
8 SET QUOTED_IDENTIFIER ON
9 GO
10
11 CREATE TABLE [dbo].[Cities](
12     [county_fips_codes] [int] NOT NULL,
13     [density] [int] NULL,
14     [city_name] [nvarchar](40) NOT NULL,
15     [state_name] [nvarchar](40) NOT NULL,
16     [latitude] [float] NOT NULL,
17     [is_military_city] [nvarchar](40) NOT NULL,
18     [longitude] [float] NOT NULL,
19     [state_abbreviation] [nvarchar](40) NOT NULL,
20     [county_name] [nvarchar](40) NOT NULL,
21     [population] [int] NOT NULL,
22     [city_id] [int] NOT NULL,
23     CONSTRAINT [PK_Cities] PRIMARY KEY CLUSTERED
24 (
25     [city_id] ASC
26 )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OP
27 ) ON [PRIMARY]
28 GO
29
30 CREATE TABLE [dbo].[OrdersFacts](
31     [order_date_id] [int] NOT NULL,
```





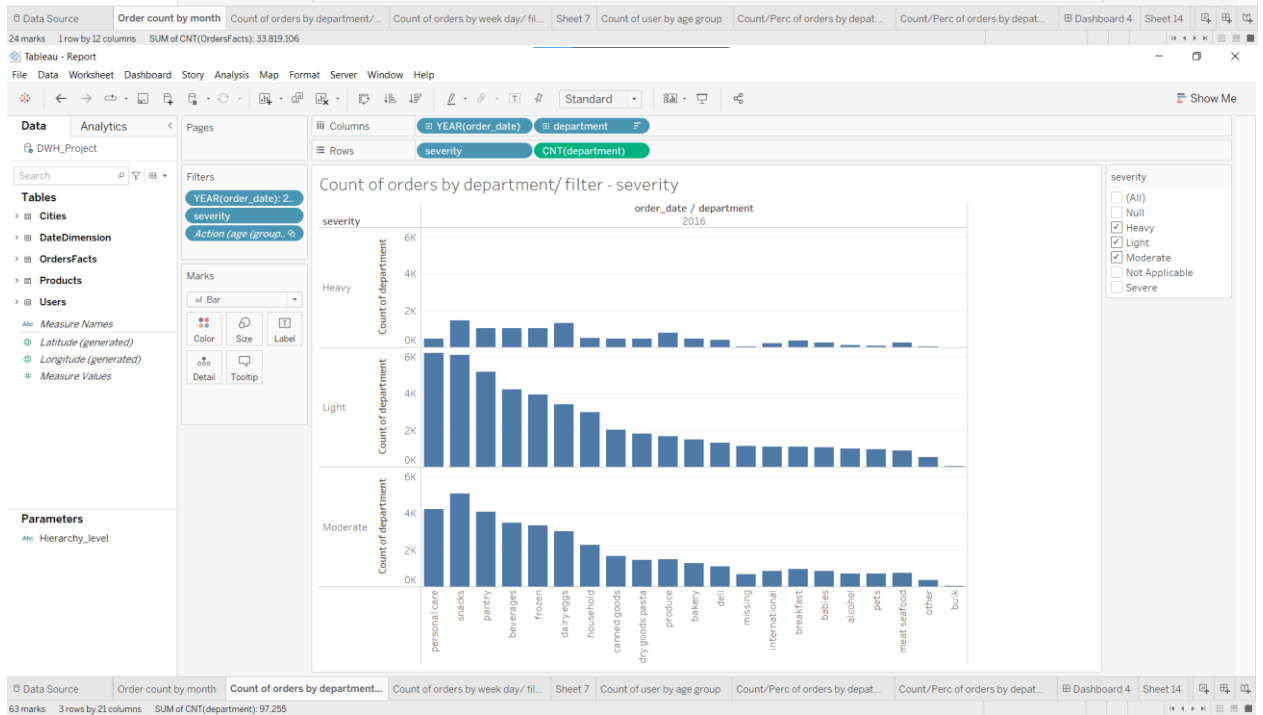
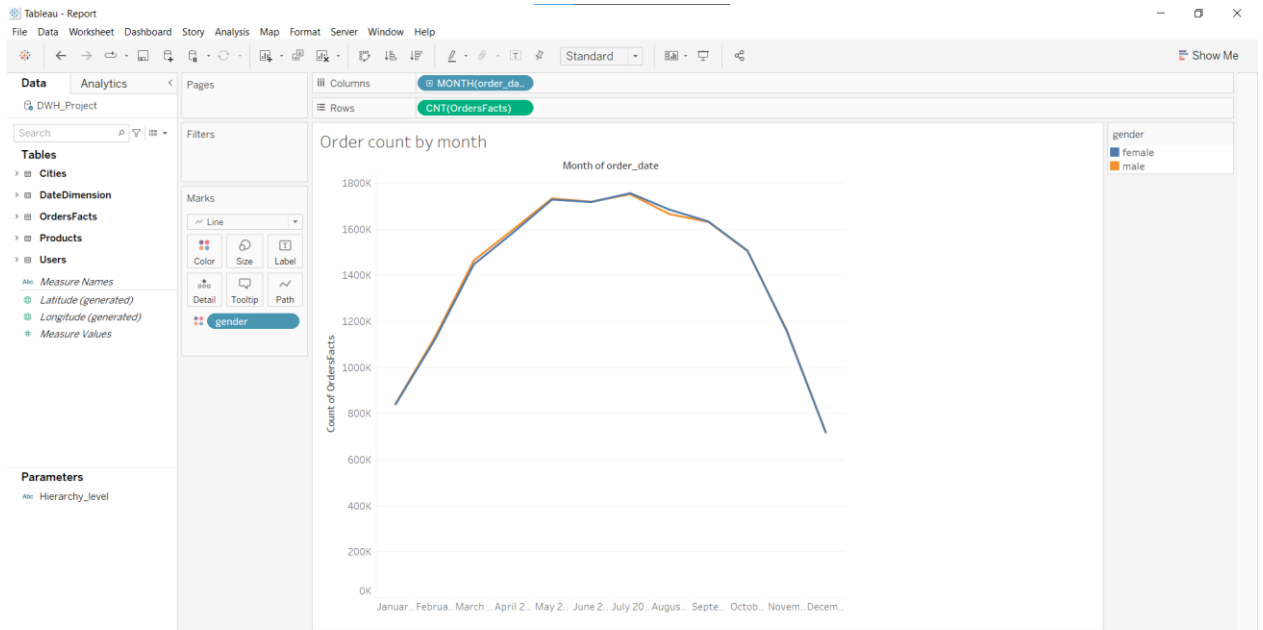
## Wczytanie Danych w systemie BI

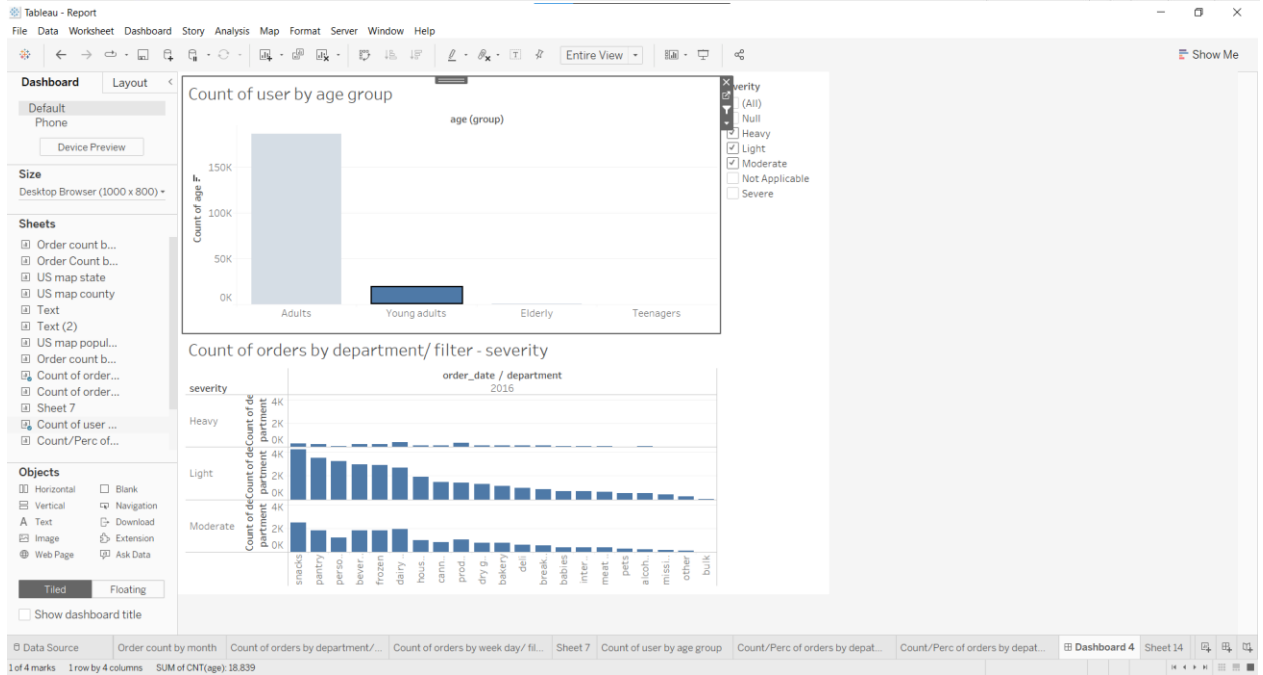
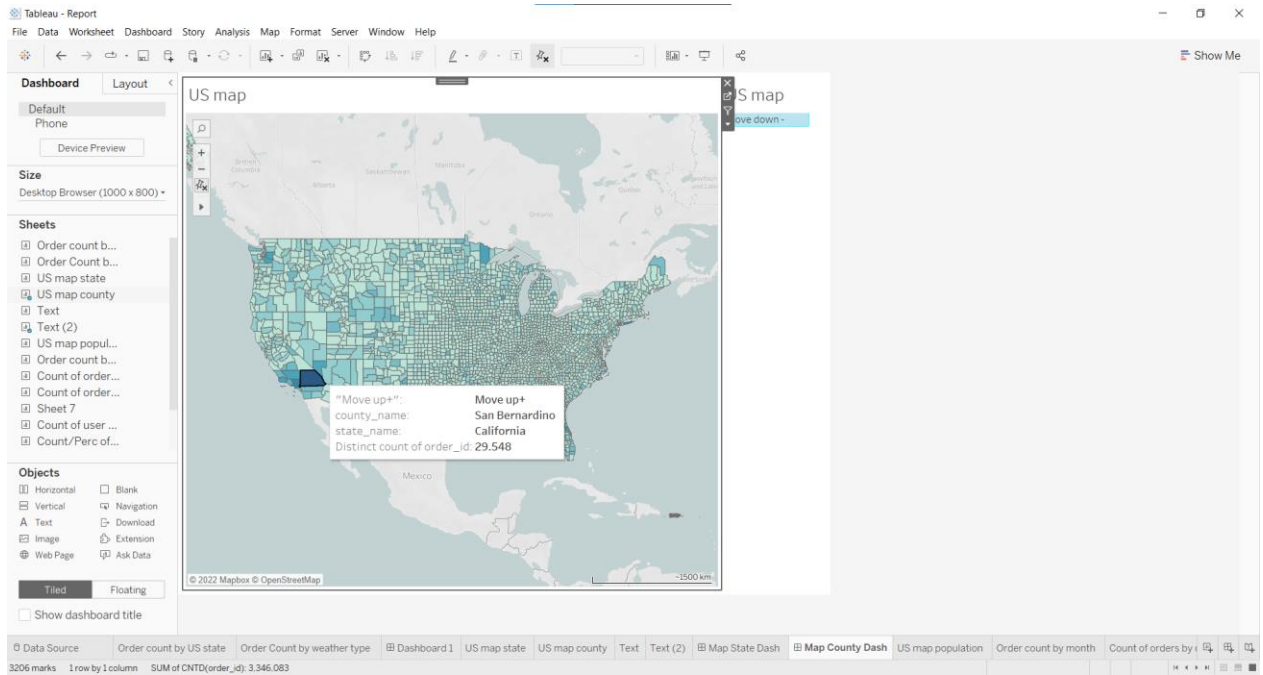
Poniżej zamieszczone jest wczytywanie danych z MS SQL Server do aplikacji BI Tableau Desktop.

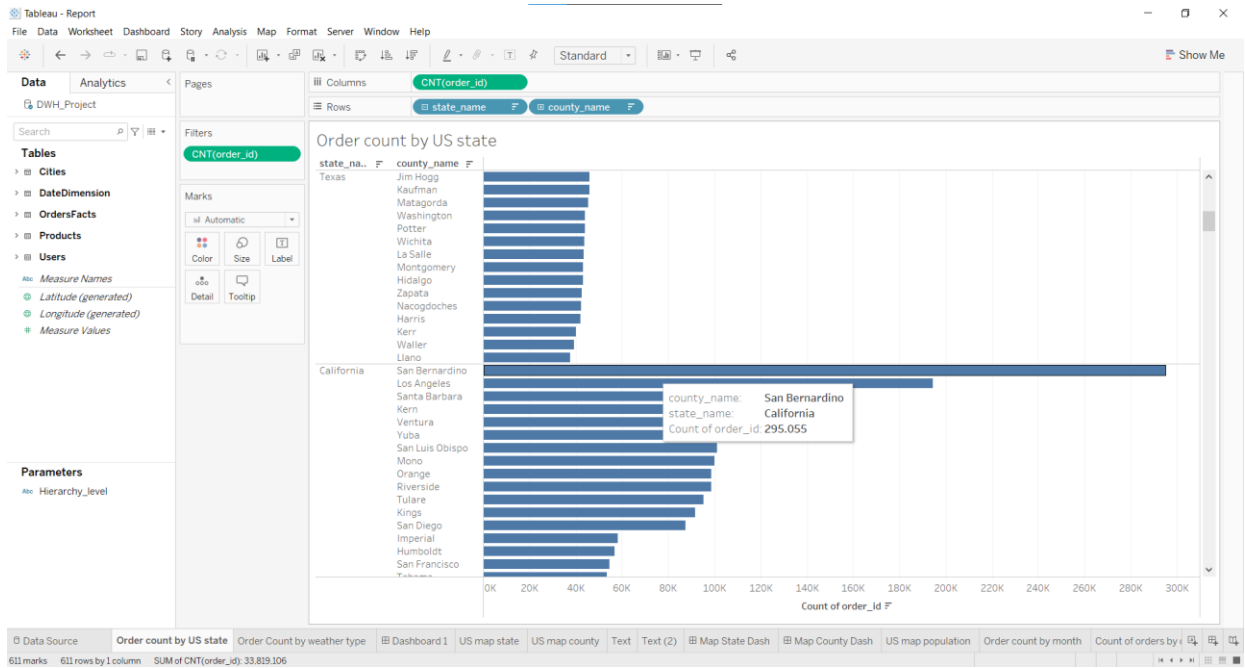


## Przykładowe wykresy i raporty

Poniżej pokazujemy część stworzonych przez nas wykresów/raportów.

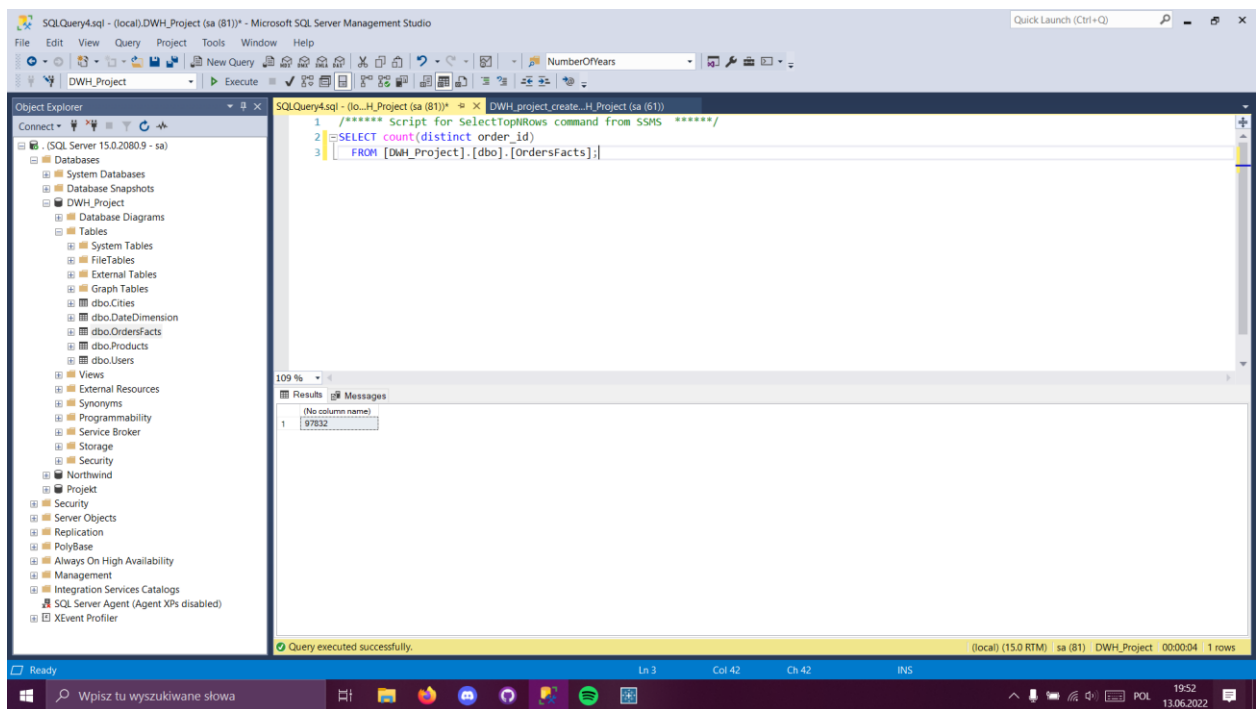


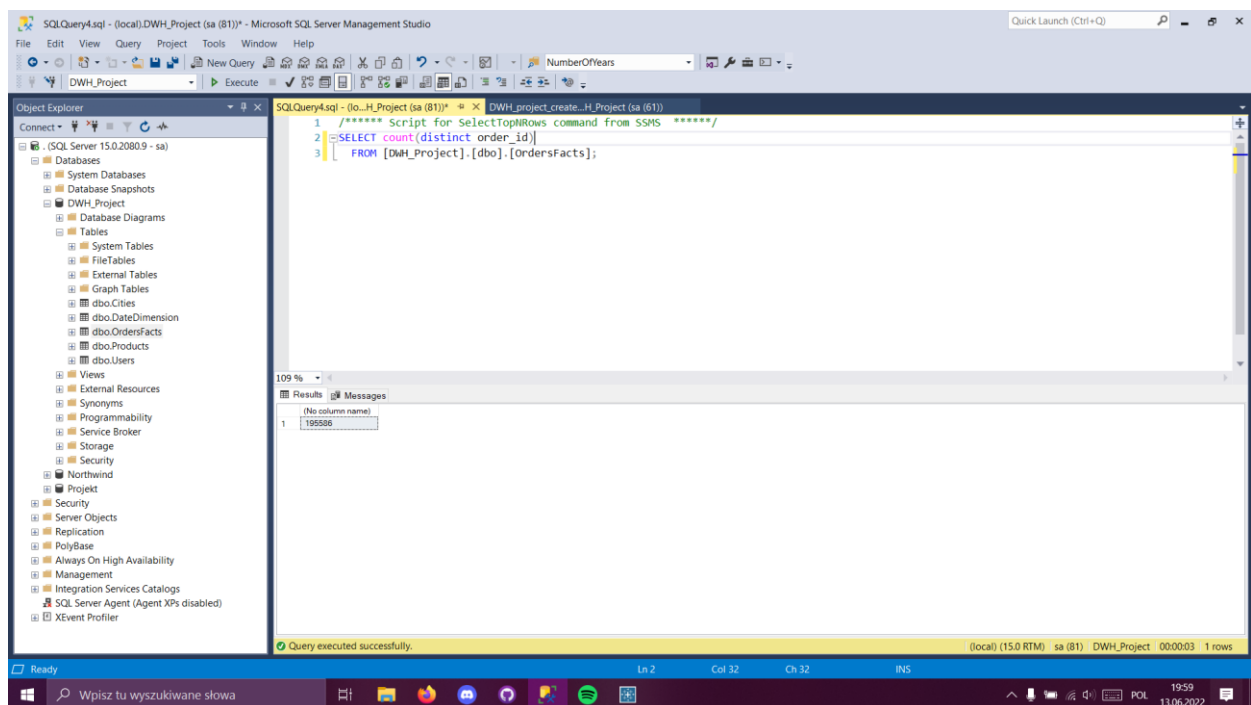
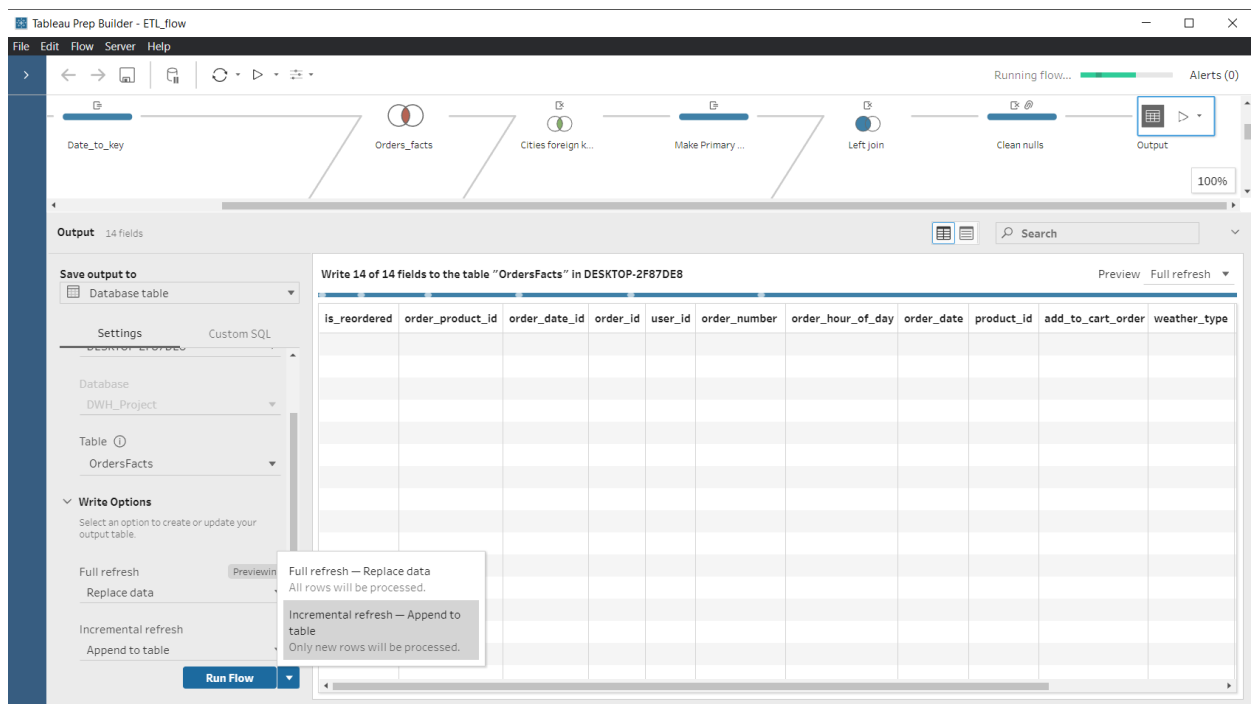




## Przykład inputacji danych:

Poniżej przedstawiamy przykład dostarczania nowych danych do hurtowni.





Tu po kolei mamy zdjęcie przed inputacją, krok inputacji i zdjęcie po inputacji danych. Widzimy że dane za pomocą incremental refresh zostały dodane do hurtowni.

## Udostępnienie raportów

Nasze raporty dostępne są dla odbiorcy systemu w formie aplikacji webowej, co istotnie ułatwia przeglądanie danych.

