

Article

[Robert Cemper](#) · Jan 2, 2022 3m read

DB Migration using SQLgateway

Thanks to [@Yuri Marx](#) we have seen a very nice example for [DB migration from Postgres to IRIS](#). My personal problem is the use of DBeaver as a migration tool. Especially as one of the strengths of IRIS (and also Caché) before is the availability of the SQLgateways that allow access to any external Db as long as for them an access using JDBC or ODBC is available. So I extended the package to demonstrate this.

It is a classic OEX package with a complete Docker. The SQLgateway is installed during Docker build and the required jdbcdriver for Linux is included in this repo In order to make this demo faster, the size of tables to migrate have been shrunk a bit.

How to test

All migration actions can be executed directly from SMP.

1. Verify the gateway connection in
[SMP> Administration> Configuration >Connectivity >SqlGatewayConfiguration](#)



To test Connection click edit

SQL Gateway Connections

Object/SQL Gateway connections provide a way to defined Gateway Connections are shown below:

Filter: Page size: 20 Items found: 1

Connection Name	DSN	User		
postgres		postgres	Edit	Delete

[Click to edit gateway connection](#)

and Test Connection

SQL Gateway Connection

Use the form below to edit a gateway connection:

Type of connection: JDBC

Connection name: postgres

User: postgres

Password:

Driver name: org.postgresql.Driver

URL: jdbc:postgresql://postgres:5432/

Class path: /opt/irisapp/jdbc/postgresql-42.3.1.jar
(May be a comma separated list if multiple jar files are required.)

Properties:

Do not use delimited identifiers by default:

Use COALESCE:

Use NVL() instead of IFNULL():

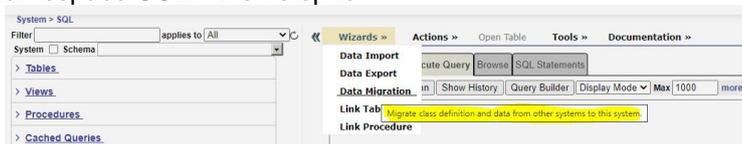
Conversion in composite Row IDs: Do not convert non-character values
 Use CAST as VARCHAR
 Use CAST as CHAR
 Use {fn convert ...}

Test Connection Save Close

Connection successful.

- verify Connection successful
- Be patient at this point. Postgres Containers sometimes take quite some time to talk to you.
- Wait a little bit, reload the page in browser and try the test again.

2. Identifying the source tables. In SMP > Change to Namespace USER then step to [SMP > Explorers > SQL > Wizards > Data Migration](#)



3. Set required import parameters

Data Migration Wizard (Namespace USER)

The Data Migration Wizard will help you copy SQL table definitions and data by using

Select a destination namespace: ▼

Schema filter: Table filter:

A percent sign (%) represents any sequence of zero or more characters.

Table type: ▼

Select a SQL Gateway connection: ▼

Schema: ▼

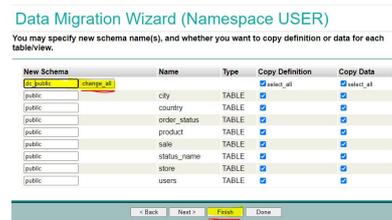
Tables: ▲

city
country
order_status
product
sale
status_name
store
users



- Destination Namespace
- Type = TABLE
- Gateway = postgres >>> now the first connection is established and you select
- Schema = public
- Tables to migrate = all

4. Identify target but change schema to be OEX compatible from public to dcpubic



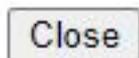
- don't forget to click change all
- we migrate Definitions and Data so both sides are selected

5. Skipping special settings we use defaults we start the task in background

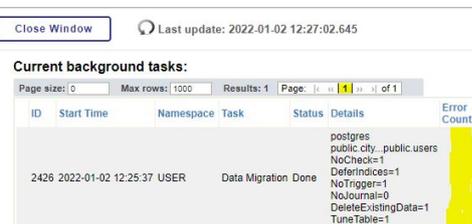
Background Job

This task will be run in the background.

[Click here to view the background tasks page.](#)

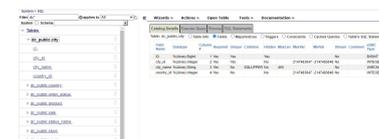


6. Now we check the results and see everything was working without Errors



- You might see errors if tables depend on the content not yet migrated.
- And wait for completions until the status shows Done

7. We terminate the Migration Wizzard and return to normal table view filtered by dc*



- All 8 tables are visible and show meaningful columns

8. Selecting a table and clicking on OpenTable shows reasonable contents

Open Table Refresh Close Window

dc_public.city in namespace USER Last update: 2022-01-02 12:41:52.574

#	ID	city_id	city_name	country_id
1	1	1	City 1	78
2	2	2	City 2	106
3	3	3	City 3	9
4	4	4	City 4	26
5	5	5	City 5	21
6	6	6	City 6	50
7	7	7	City 7	65
8	8	8	City 8	80
9	9	9	City 9	17
10	10	10	City 10	68
11	11	11	City 11	43
12	12	12	City 12	83
13	13	13	City 13	6
14	14	14	City 14	17
15	15	15	City 15	64
16	16	16	City 16	79
17	17	17	City 17	2
18	18	18	City 18	64
19	19	19	City 19	10
20	20	20	City 20	38
21	21	21	City 21	30
22	22	22	City 22	54
23	23	23	City 23	60
24	24	24	City 24	48
25	25	25	City 25	31
26	26	26	City 26	81
27	27	27	City 27	64
28	28	28	City 28	78
29	29	29	City 29	34
30	30	30	City 30	29

Complete

Open Table Refresh Close Window

dc_public.sale in namespace USER

Last update: 2022-01-02 12:57:18.116

#	ID	sale_id	amount	date_sale	product_id	user_id	store_id	
1	2	a0761ca1-719b-4bc0-9a76-4f2c310e5b65	3249.345	06/05/2019 20:18:06.214735		19	227173	360
2	3	5fe0daea-1fcc-4af7-87a9-389e548cdc6c	9610.107	08/31/2019 08:13:15.672707		262	106284	303
3	4	51fa9ce9-b9a6-48de-ab84-4ea9104d747b	8605.735	01/31/2019 16:30:04.2781		117	25490	422
4	5	f64395f3-6f3e-42c0-9750-13b52d8e9d91	5340.493	05/03/2019 07:50:09.398723		8	60640	363
5	6	d5ee34eb-e6b7-4364-b13e-bab0fc56774e	6803.837	11/15/2019 14:19:24.413511		274	225373	106
6	7	ea4c67ff-6111-49b1-bfdb-27a349e8de2c	8675.357	03/28/2019 07:31:38.920049		234	52221	156
7	8	6ad4b406-6135-4a6e-aabc-c5a93021dc76	7329.581	06/13/2019 09:01:39.499335		163	95381	404
8	9	1d5ec4fa-d78d-4959-8002-8ec32bb220e3	2608.777	04/08/2019 19:33:23.187622		58	43114	457
9	10	a0be6eb1-5e7f-48b0-82c4-987a3bb2cea3	2244.691	10/22/2019 21:27:37.973024		174	130126	126
10	11	4da07a5f-8706-4f2f-8c7b-1e86ee7ed562	2980.329	05/25/2019 17:24:58.376043		276	197371	368
11	12	f597cfef-e27c-4d15-82e5-eff923e4fe56	5321.619	08/02/2019 18:19:53.805683		89	114176	498
12	13	e1775f83-40cf-493e-ad97-c31569b7a3de	2141.334	12/27/2019 05:21:23.834163		102	68660	262
13	14	93d7eaac-9f91-429e-ae7-4badf24ad4d3	7774.936	09/19/2019 12:13:15.479316		200	177053	104
14	15	c5c92f0b-f057-4e35-a69e-f40a010fa4f2	5616.166	09/20/2019 22:29:13.846402		56	65308	296
15	16	b7a71959-e6c8-4fd3-b33e-89206cec6f79	7224.649	05/03/2019 20:19:54.467358		196	97730	419
16	17	59b158ed-a3c0-4846-941c-d4ca9d8554cc	3970.118	11/05/2019 08:45:04.906936		237	235906	161
17	18	d0f9dd93-4dcf-4d3e-bc10-84f5f9e07d4b	3233.518	09/07/2019 03:53:03.221553		271	19634	467
18	19	51bec7e8-2402-4e35-9f72-70baaa8c5e76	3444.624	04/10/2019 11:04:22.86936		6	155089	145
19	20	50f415ea-f903-4816-9559-22770e4dda28	9268.943	01/03/2019 14:34:58.248772		140	197052	151
20	21	05f6076f-bfb2-4ae4-97b2-000ade0871e4	137.647	02/14/2019 17:46:34.40058		288	25637	188
21	22	10997c1c-9dc7-40ce-b850-1b354a376f2b	1973.726	11/22/2019 07:26:28.664947		220	94132	478
22	23	41b6bc50-043c-4c39-b22e-8bc02e13f852	2474.278	12/06/2019 05:10:50.654773		16	152178	103
23	24	2466ed9e-18bd-45e4-86dc-31057656de4f	1172.654	11/09/2019 10:20:55.203226		285	20773	257
24	25	51ea7404-3daf-43d5-a076-7fe189e4b4c2	8668.224	09/17/2019 11:13:35.441098		284	38822	343

9.

A look into the related generated Class Definitions confirms the result and successful completion.

