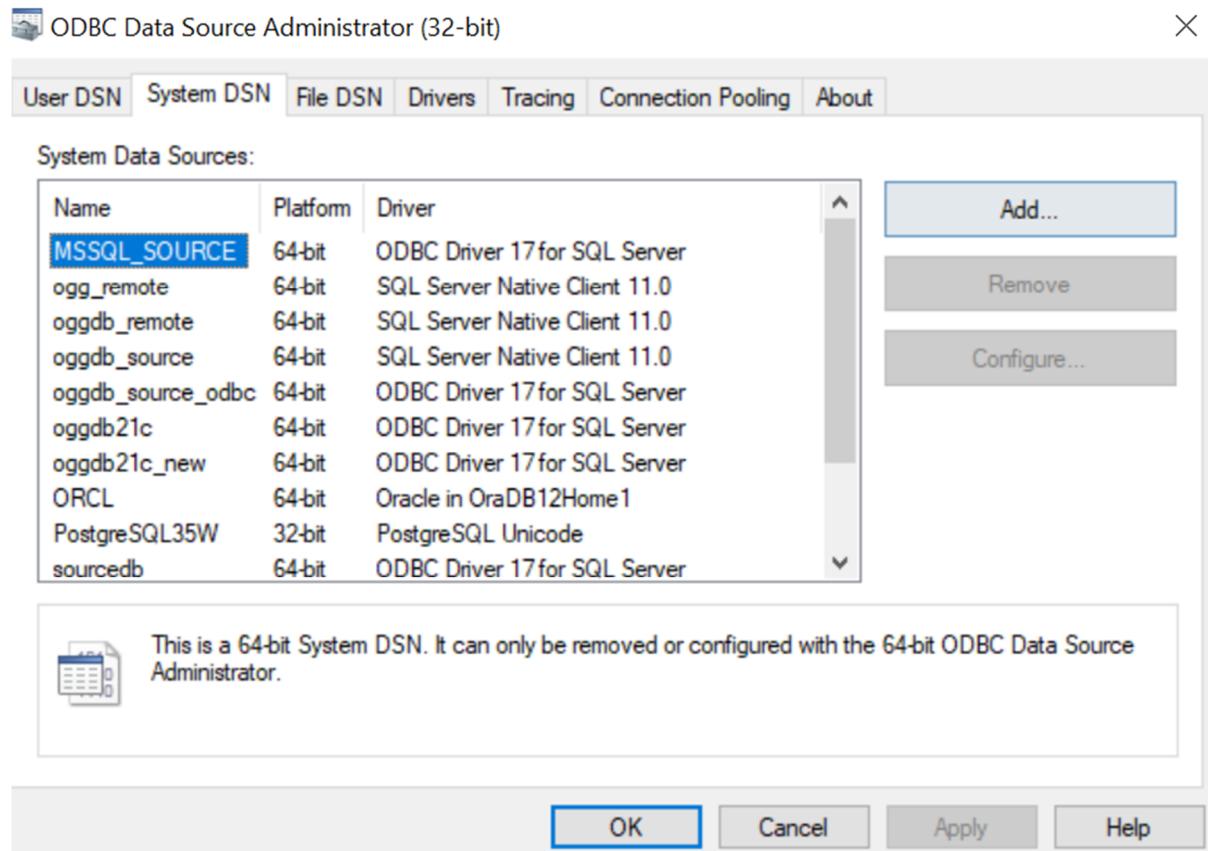


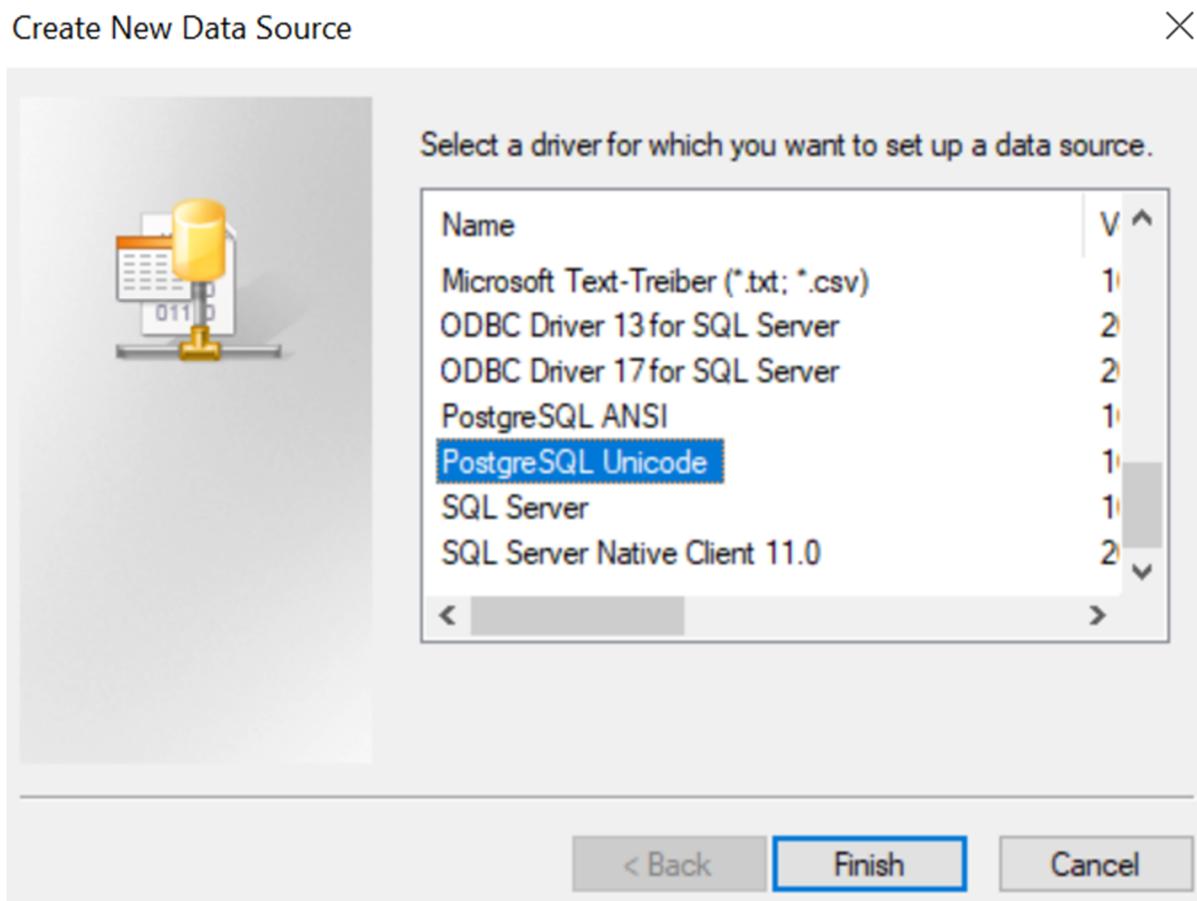
## MS SQL Server to PostgreSQL Data Migration using MS SQL Export Data

### Install PostgreSQL ODBC driver and create system DSN for PSQL database

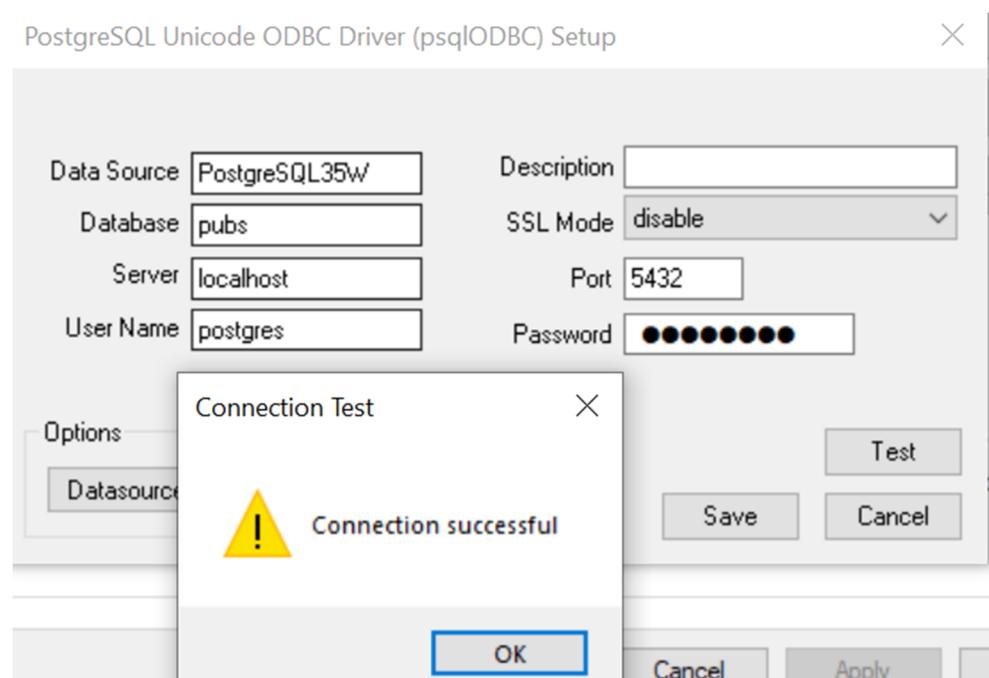
Click Add



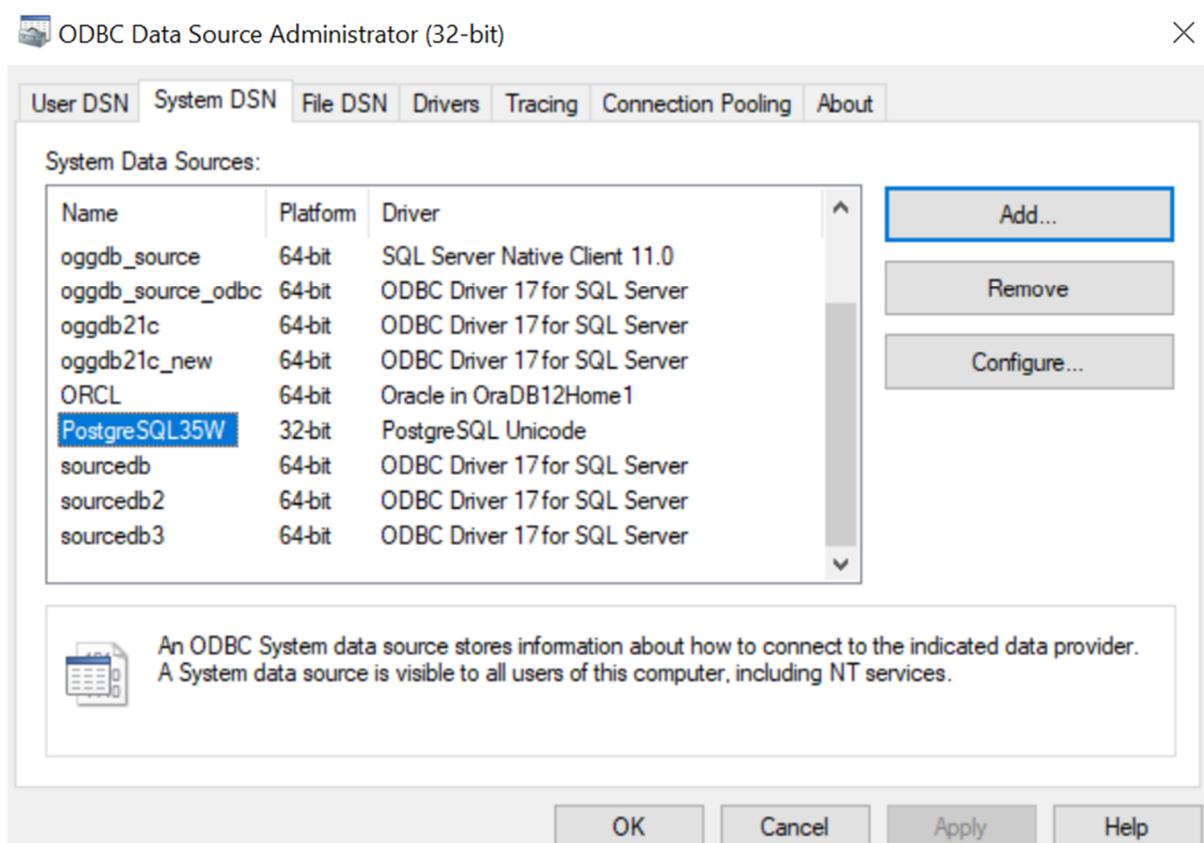
**Choose PostgreSQL Unicode > click Finsh**



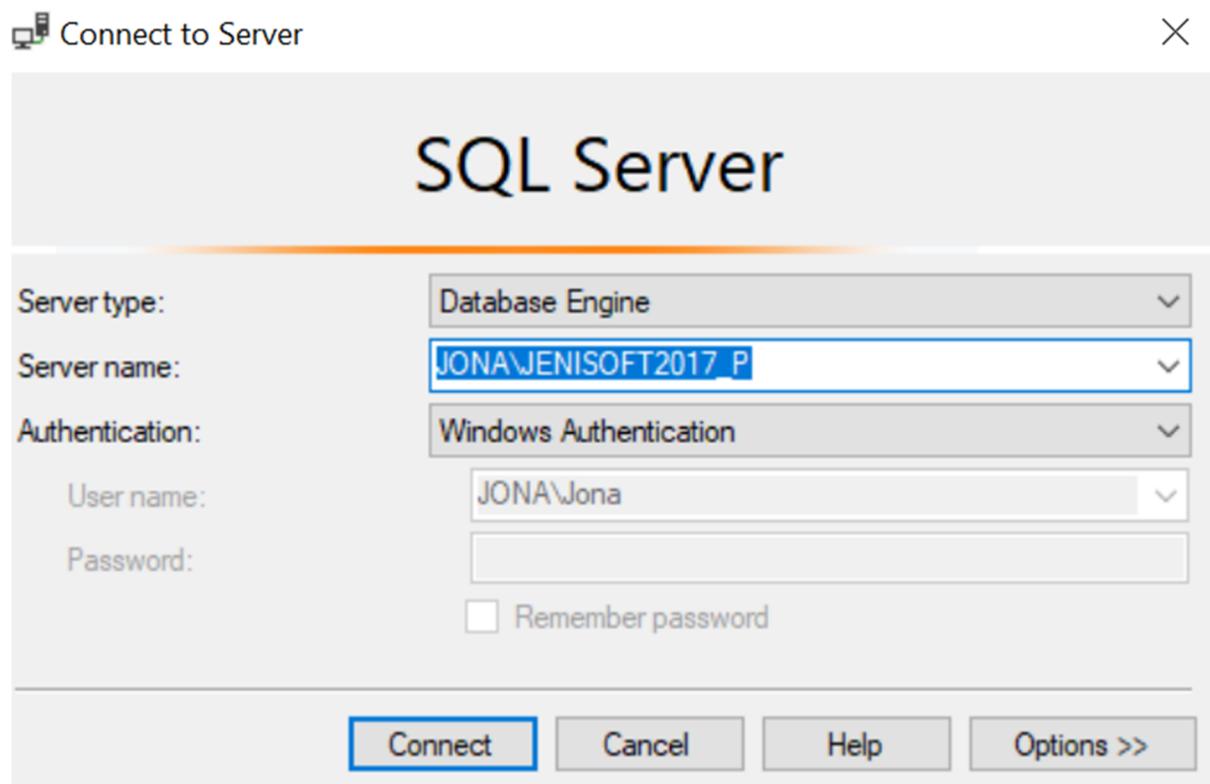
**Provide require Details and then Click Test > Save**



**Click OK**



Connect MSSQL Server:



Source database pubs:

The screenshot shows the Object Explorer window in SQL Server Management Studio. The connection is to JONA\JENISOFT2017\_P (SQL Server 14.0.2052.1 - JONA\Jona). The 'Databases' node is expanded, showing System Databases, Database Snapshots, PRODUCT DATABASE, and the selected 'pubs' database. The 'Tables' node under 'pubs' is also expanded, listing various system and user tables: authors, discounts, employee, jobs, pub\_info, publishers, roysched, sales, stores, titleauthor, and titles. Other nodes like Views, External Resources, Synonyms, Programmability, and Service Broker are also visible.

- Object Explorer ▾ X
- Connect ▾
- JONA\JENISOFT2017\_P (SQL Server 14.0.2052.1 - JONA\Jona)
- Databases
  - + System Databases
  - + Database Snapshots
  - + PRODUCT DATABASE
  - + pubs
    - + Database Diagrams
    - + Tables
      - + System Tables
      - + FileTables
      - + External Tables
      - + Graph Tables
      - + dbo.authors
      - + dbo.discounts
      - + dbo.employee
      - + dbo.jobs
      - + dbo.pub\_info
      - + dbo.publishers
      - + dbo.roysched
      - + dbo.sales
      - + dbo.stores
      - + dbo.titleauthor
      - + dbo.titles
    - + Views
    - + External Resources
    - + Synonyms
    - + Programmability
    - + Service Broker

Connect PGAdmin and create empty pubs database:

The screenshot shows the pgAdmin interface. In the left sidebar, under 'Servers', there are nine entries: AWS-RDS-PG, Linux-PSQL14, Linux\_PSQL14\_NEW, PSQL\_Linux\_5433, PostgreSQL 12, PostgreSQL 13, and PostgreSQL 14. Under 'PostgreSQL 14', the 'Databases' section is expanded, showing 'postgres', 'Login/Groups', and 'Tablespace'. A context menu is open over the 'Databases' entry, with options 'Create' and 'Database...'. Below this, a 'Create - Database' dialog box is open. The 'General' tab is selected, showing 'Database' set to 'pubs' and 'Owner' set to 'postgres'. At the bottom of the dialog are buttons for 'Cancel', 'Reset', and 'Save'.

pgAdmin

File ▾ Object ▾ Tools ▾ Help ▾

Browser Dashboard Properties

Servers (9)

- AWS-RDS-PG
- Linux-PSQL14
- Linux\_PSQL14\_NEW
- PSQL\_Linux\_5433
- PostgreSQL 12
- PostgreSQL 13
- PostgreSQL 14
  - Databases
    - Create
    - Database...
    - Refresh...
  - Login/Groups
  - Tablespace

Create - Database

General Definition Security Parameters SQL

Database: pubs

Owner: postgres

Comment:

i ? Cancel Reset Save

Create dbo schema on pubs database on PG:

The screenshot shows the pgAdmin interface for PostgreSQL 14. On the left, a tree view displays the database structure under 'PostgreSQL 14' > 'Databases' (2) > 'pubs'. A context menu is open over the 'Schemas (1)' entry, with 'Create' highlighted. Below this, a modal window titled 'Create - Schema' is open, showing the 'General' tab selected. The 'Name' field contains 'dbo' and the 'Owner' field is set to 'postgres'. At the bottom of the modal are buttons for 'Cancel', 'Reset', and 'Save'.

General Security Default privileges SQL

Name: dbo

Owner: postgres

Comment:

**i** **?** Cancel Reset Save

Currently no tables on dbo schema on PG

The screenshot shows a tree view of database objects. At the top level, there is a node for 'Schemas (2)'. Below it, the 'dbo' schema is expanded, showing the following objects:

- > Collations
- > Domains
- > FTS Configurations
- > FTS Dictionaries
- > FTS Parsers
- > FTS Templates
- > Foreign Tables
- > Functions
- > Materialized Views
- > Procedures
- > Sequences
- > Tables
- > Trigger Functions
- > Types
- > Views

At the bottom of the list, there is another node for 'public'.

## Create user defined data type on PG:

The screenshot shows the pgAdmin 4 interface. The left sidebar is titled 'Browser' and lists various database objects under 'Schemas (2)'. One of the schemas, 'dbo', is expanded, showing sub-objects like Collations, Domains, FTS Configurations, etc. The main area is a 'Query Editor' window with the following SQL code:

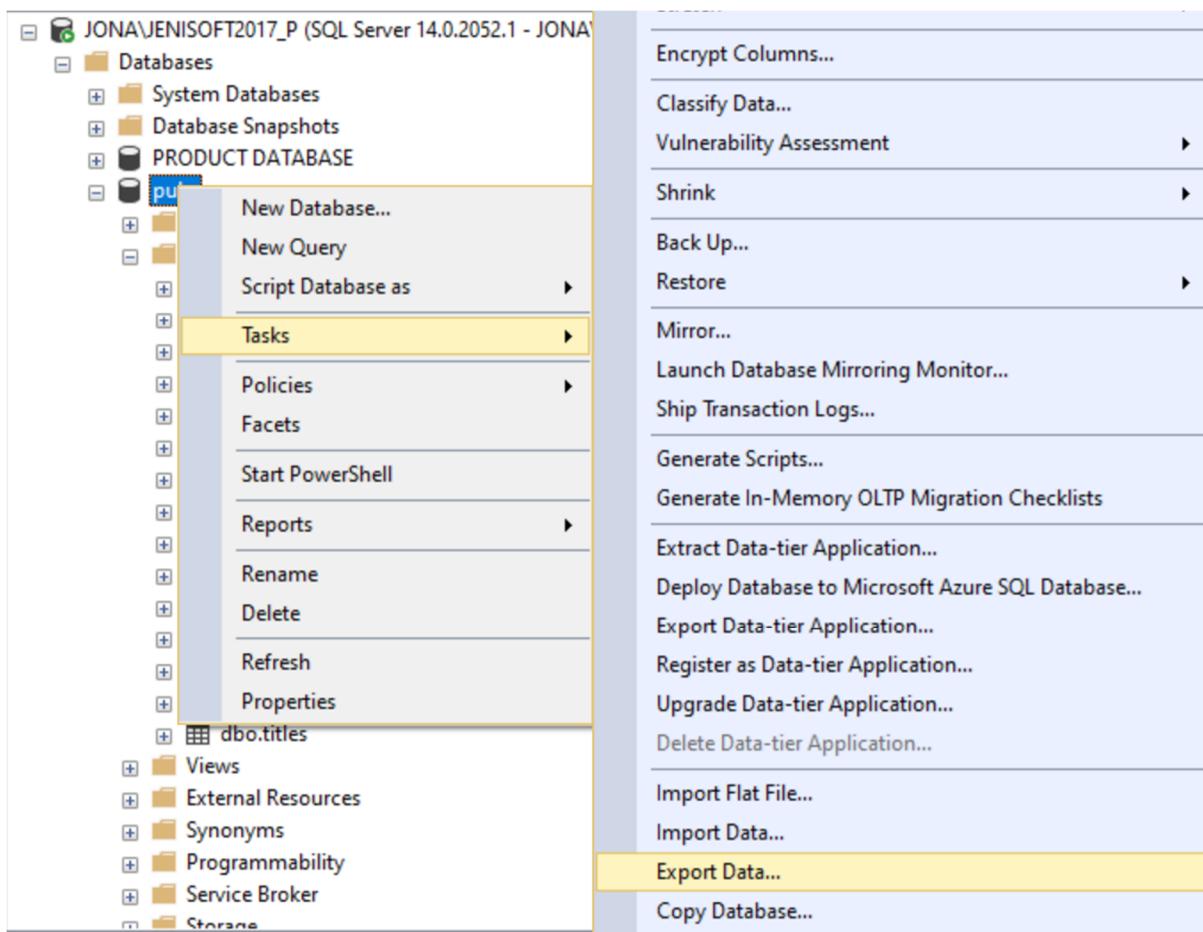
```
1 CREATE TYPE dbo.empid AS (empid char(9));
2
3 CREATE TYPE dbo.id AS (id varchar(11));
4
5 CREATE TYPE dbo.tid as (tid varchar(6));
```

Below the code, the 'Messages' panel displays the output:

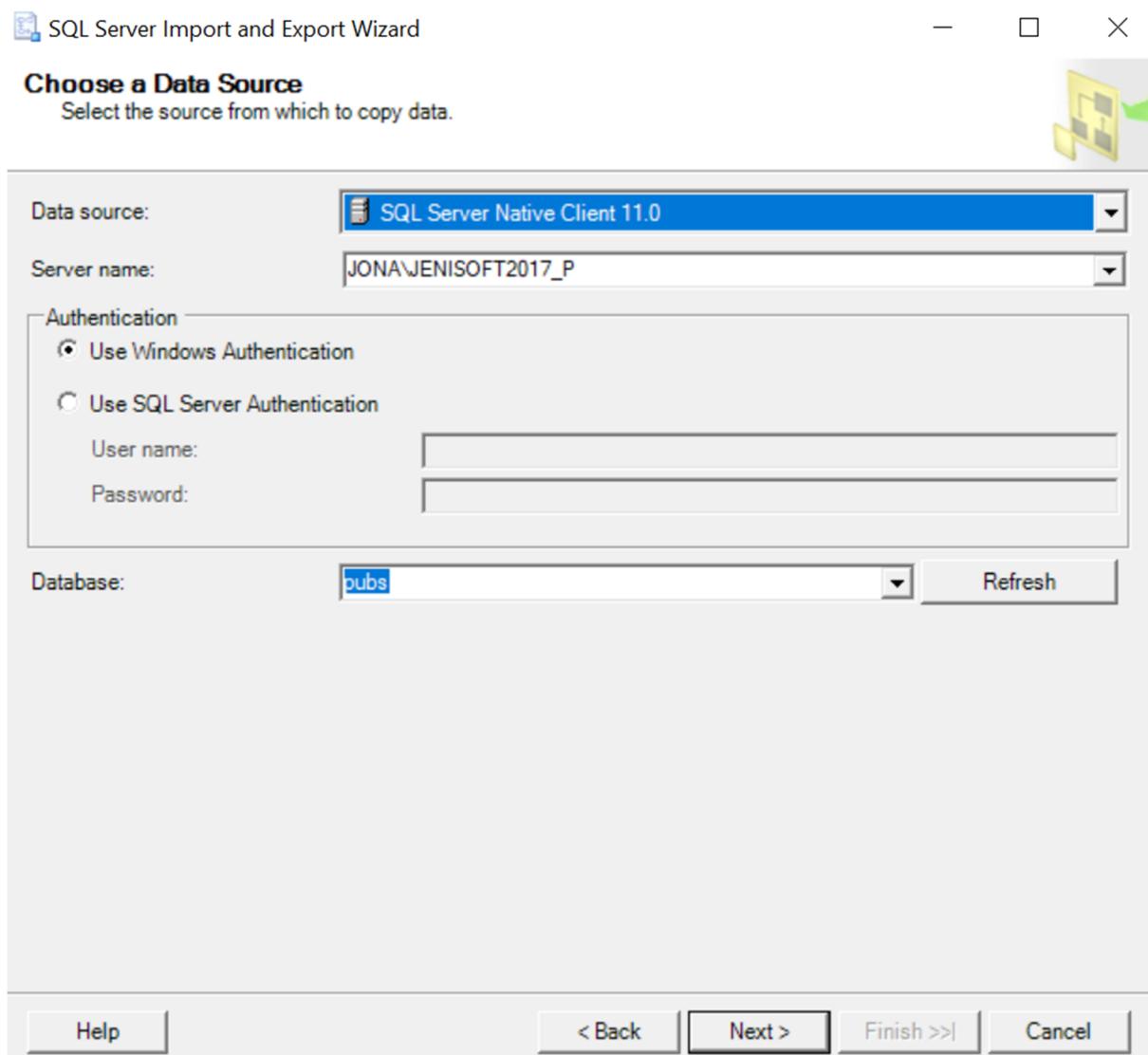
CREATE TYPE  
Query returned successfully in 83 msec.

## Migrated MSSQL Pubs database tables to PostgreSQL Pubs database

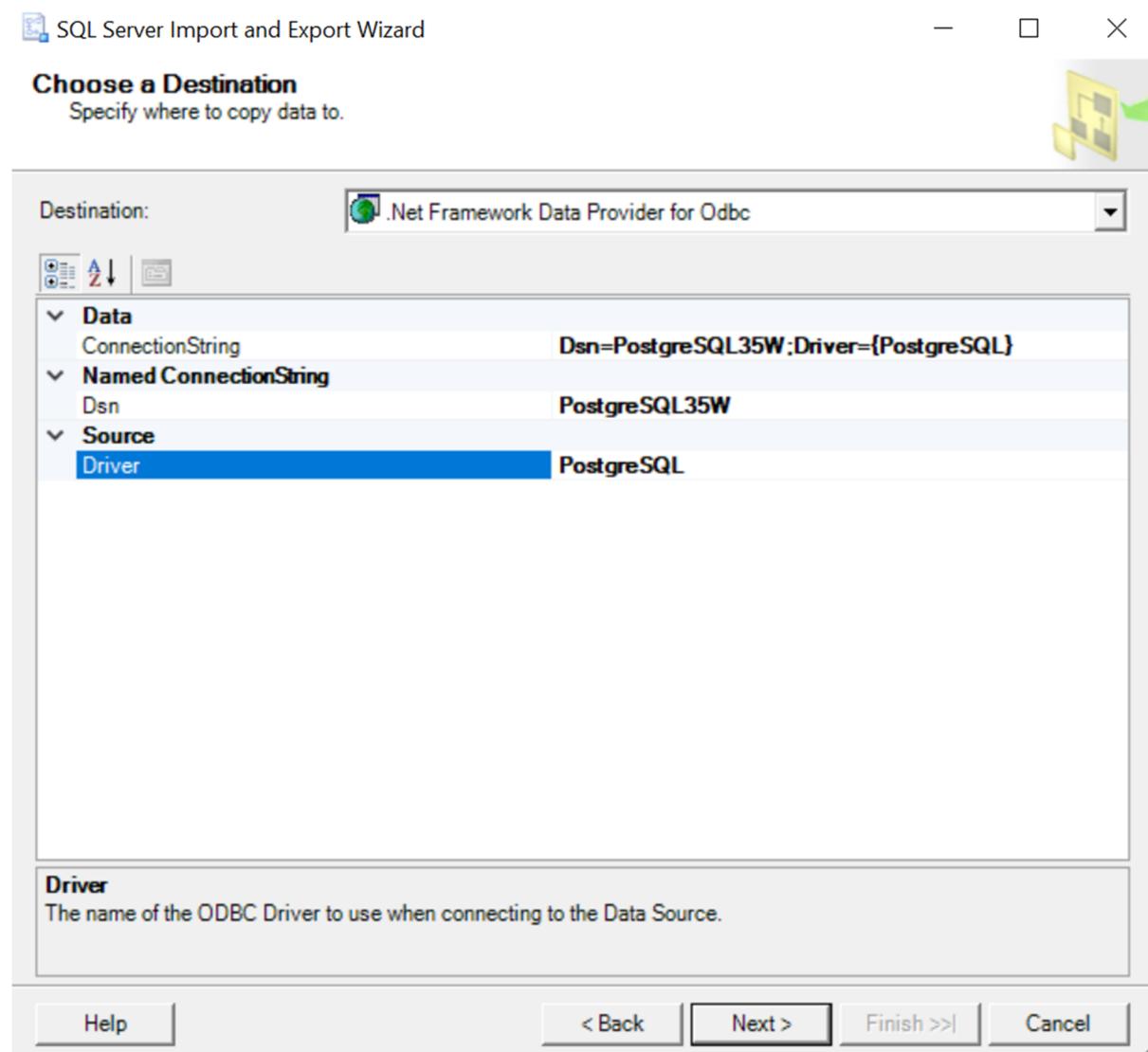
Right Click Pubs database > Tasks > Export Data



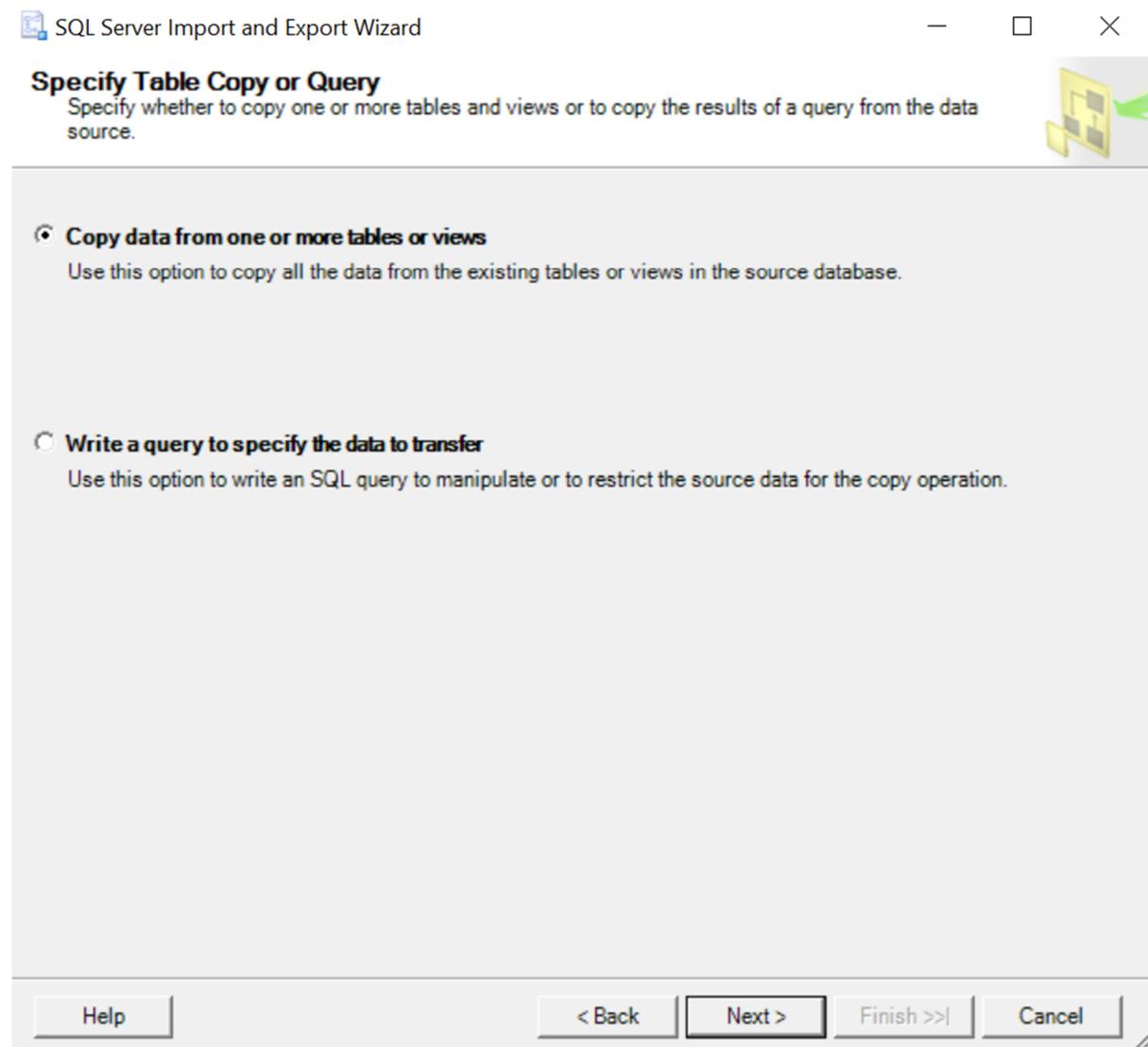
## Choose Source Database data source



## Type Destination database DSN and Driver



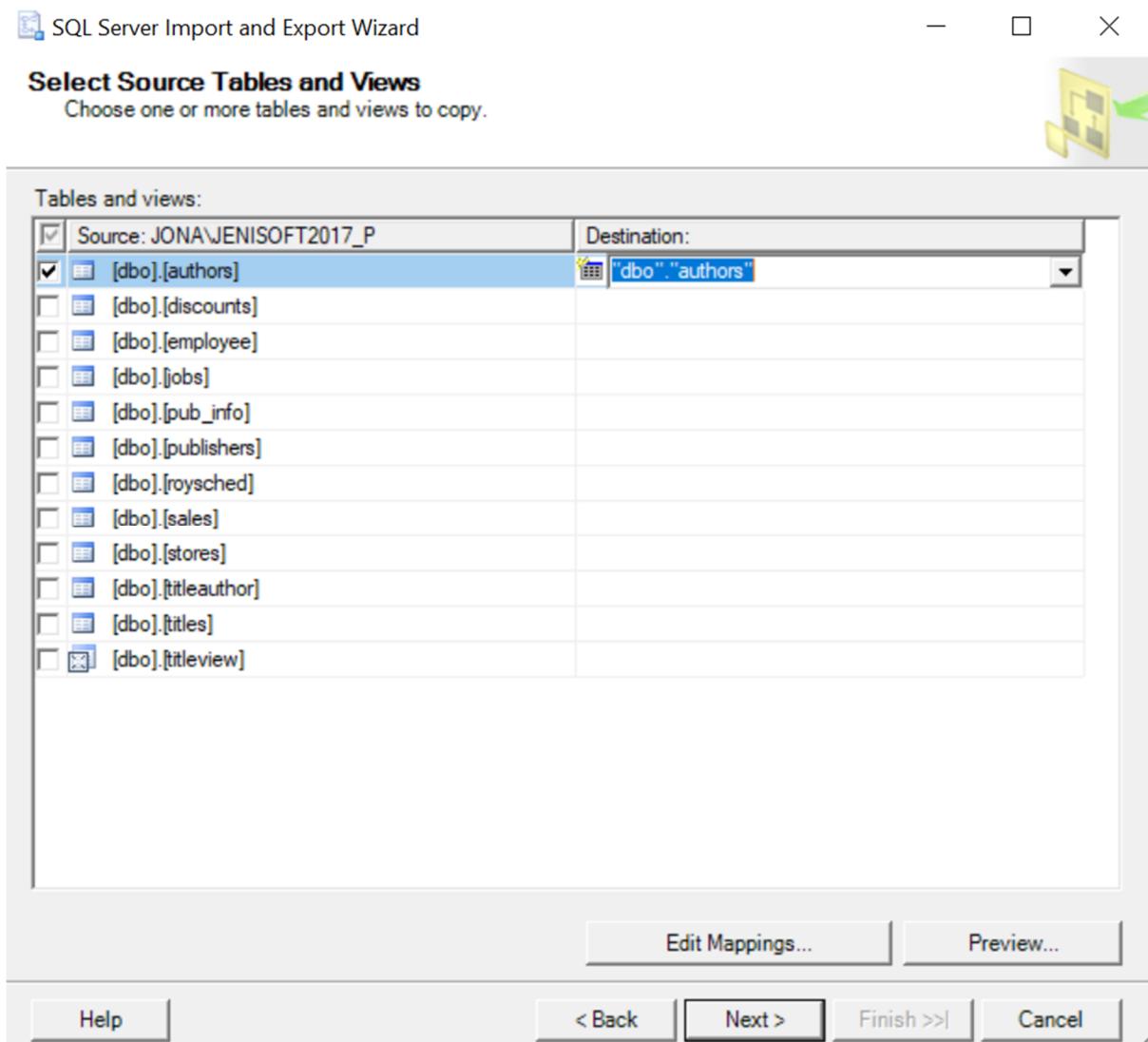
Choose “Copy data from one or more tables or views”



## Select Source database's table and type destination database table name

- Here we are moving data from MSSQL dbo schema authors tables to PG dbo schema authors schema.

### Click Edit Mapping



**Click Edit SQL**

Column Mappings

Source: [dbo].[authors]

Destination: "dbo"."authors"

Create destination table

Delete rows in destination table  Drop and re-create destination table

Append rows to the destination table

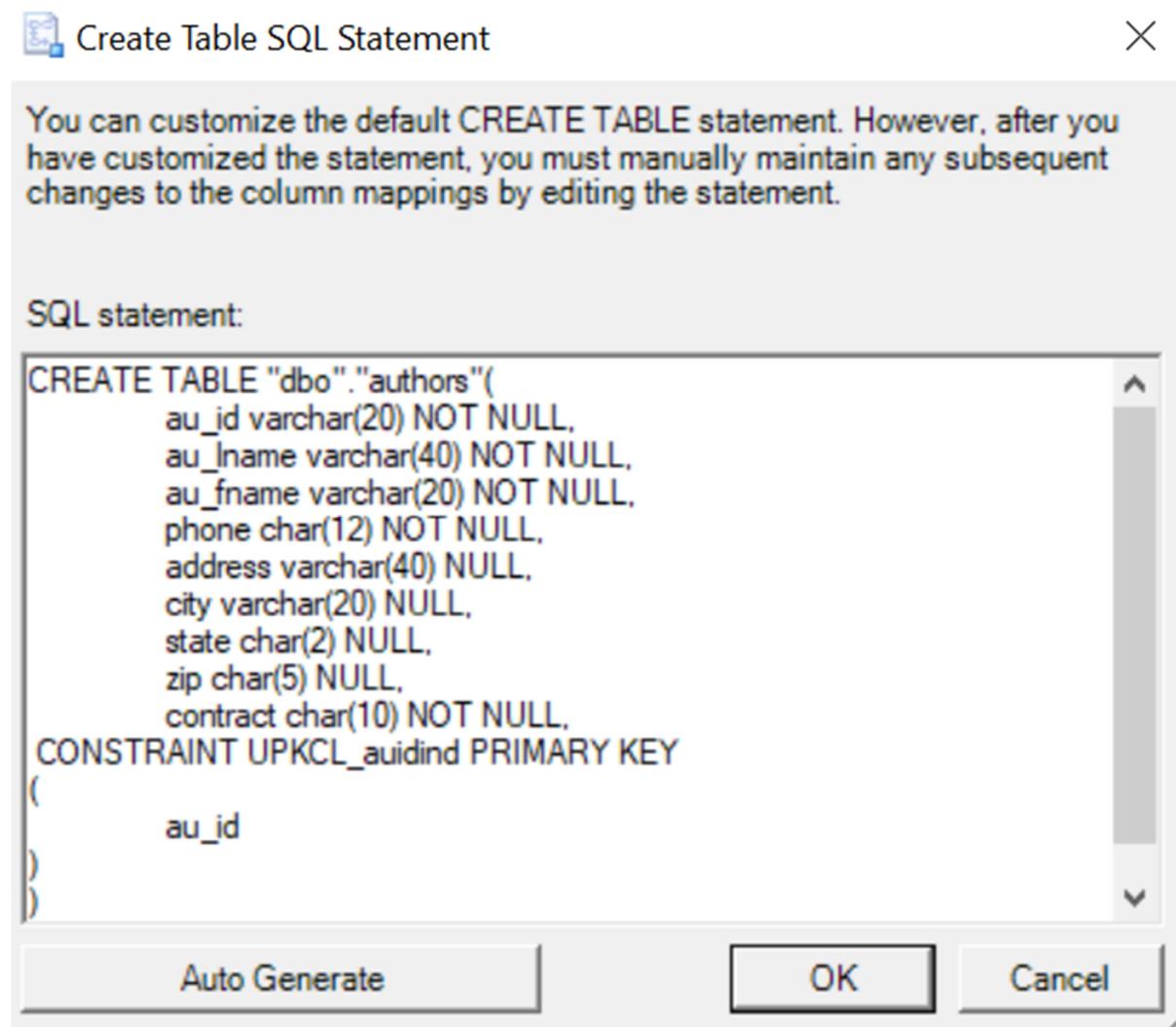
Mappings:

Source	Destination	Type	Nullable	Size	Precision	Scale
au_id	au_id	varchar	<input type="checkbox"/>			
au_lname	au_lname	varchar	<input type="checkbox"/>			
au_fname	au_fname	varchar	<input type="checkbox"/>			
phone	phone	char	<input type="checkbox"/>			
address	address	varchar	<input checked="" type="checkbox"/>			
city	city	varchar	<input checked="" type="checkbox"/>			
state	state	char	<input checked="" type="checkbox"/>			
zip	zip	char	<input checked="" type="checkbox"/>			
contract	contract	bool	<input type="checkbox"/>			

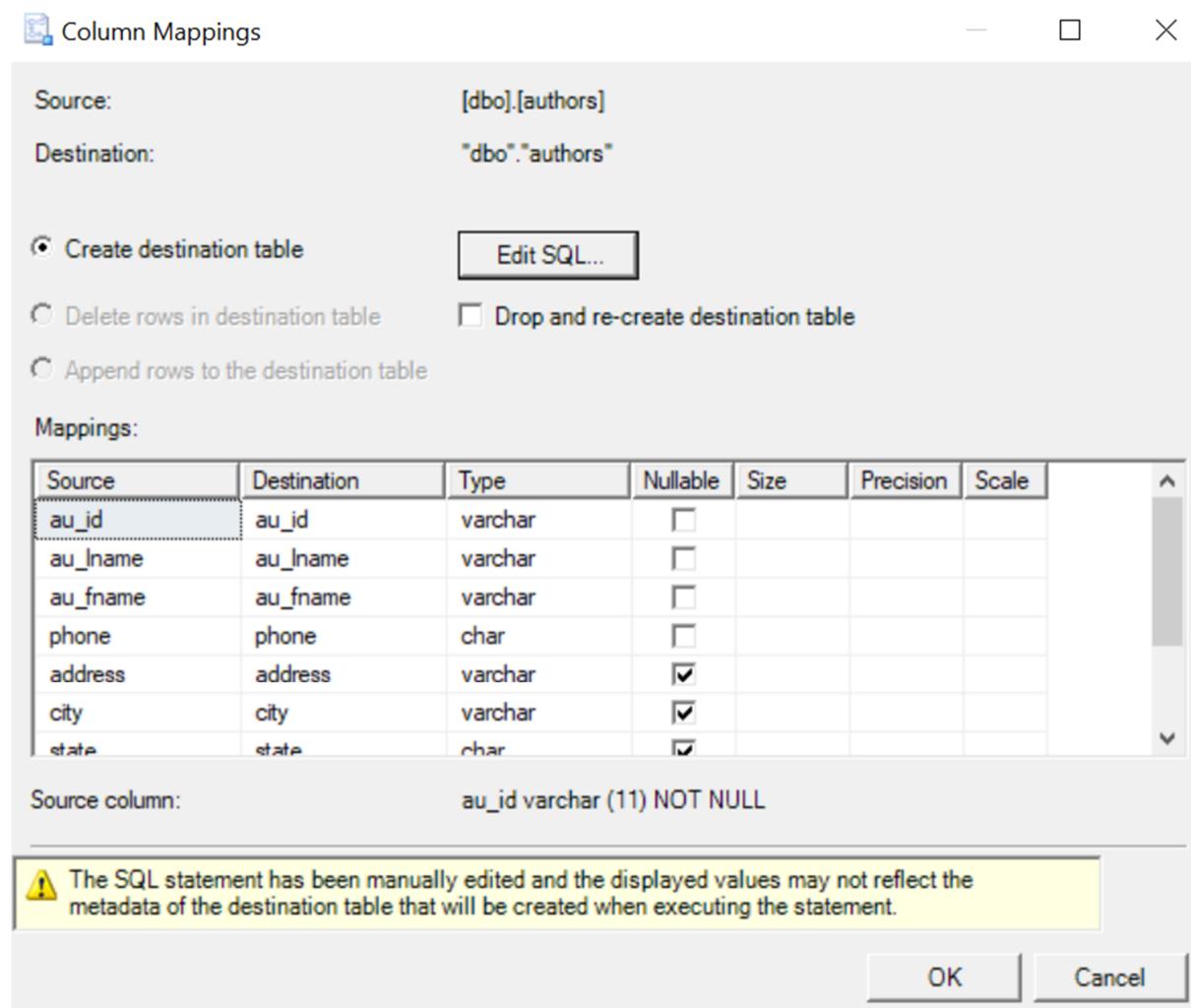
Source column: au\_id varchar (11) NOT NULL

Type converted PSQL Table creation DDL statements

Click Ok

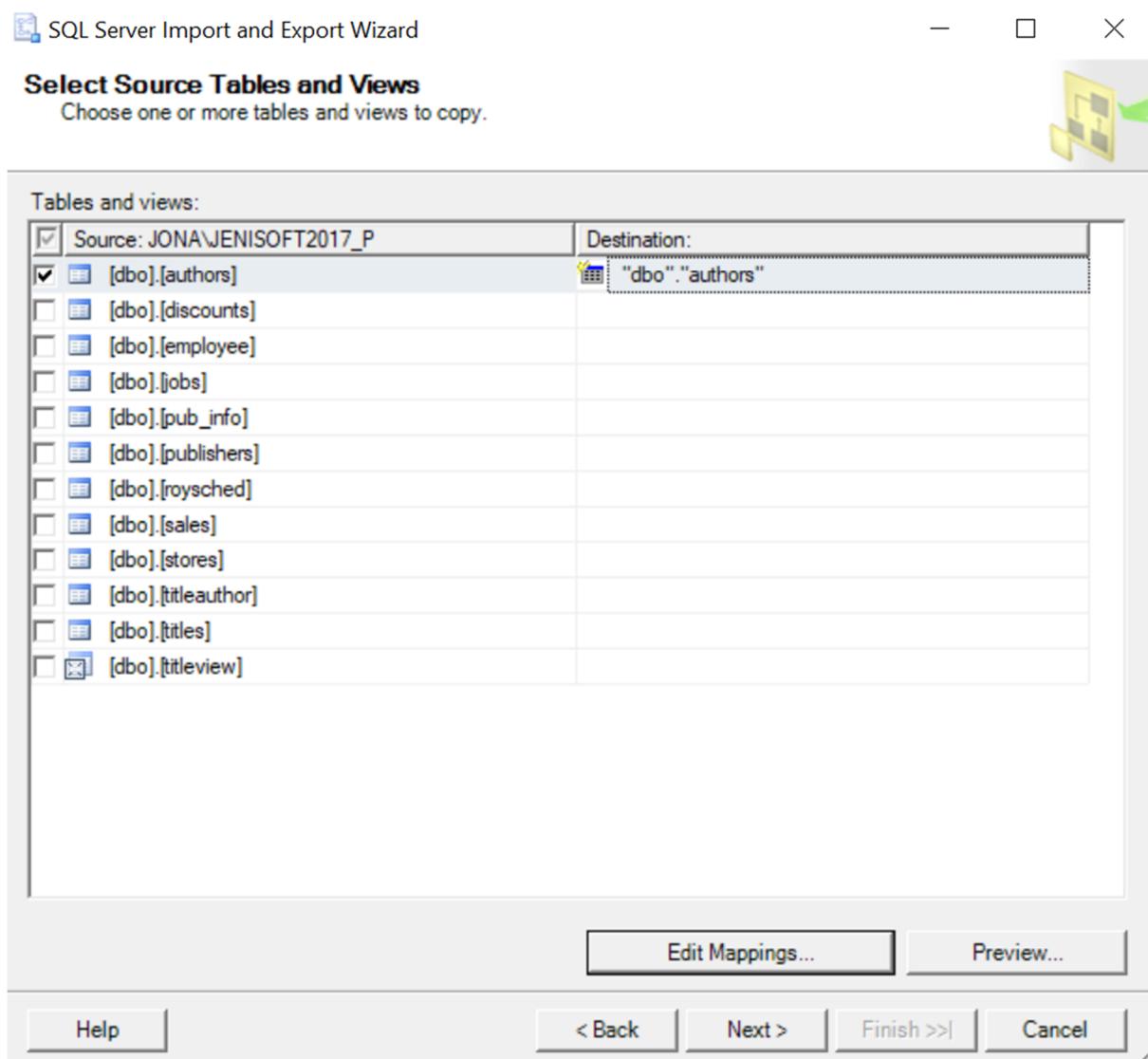


**Click Ok**

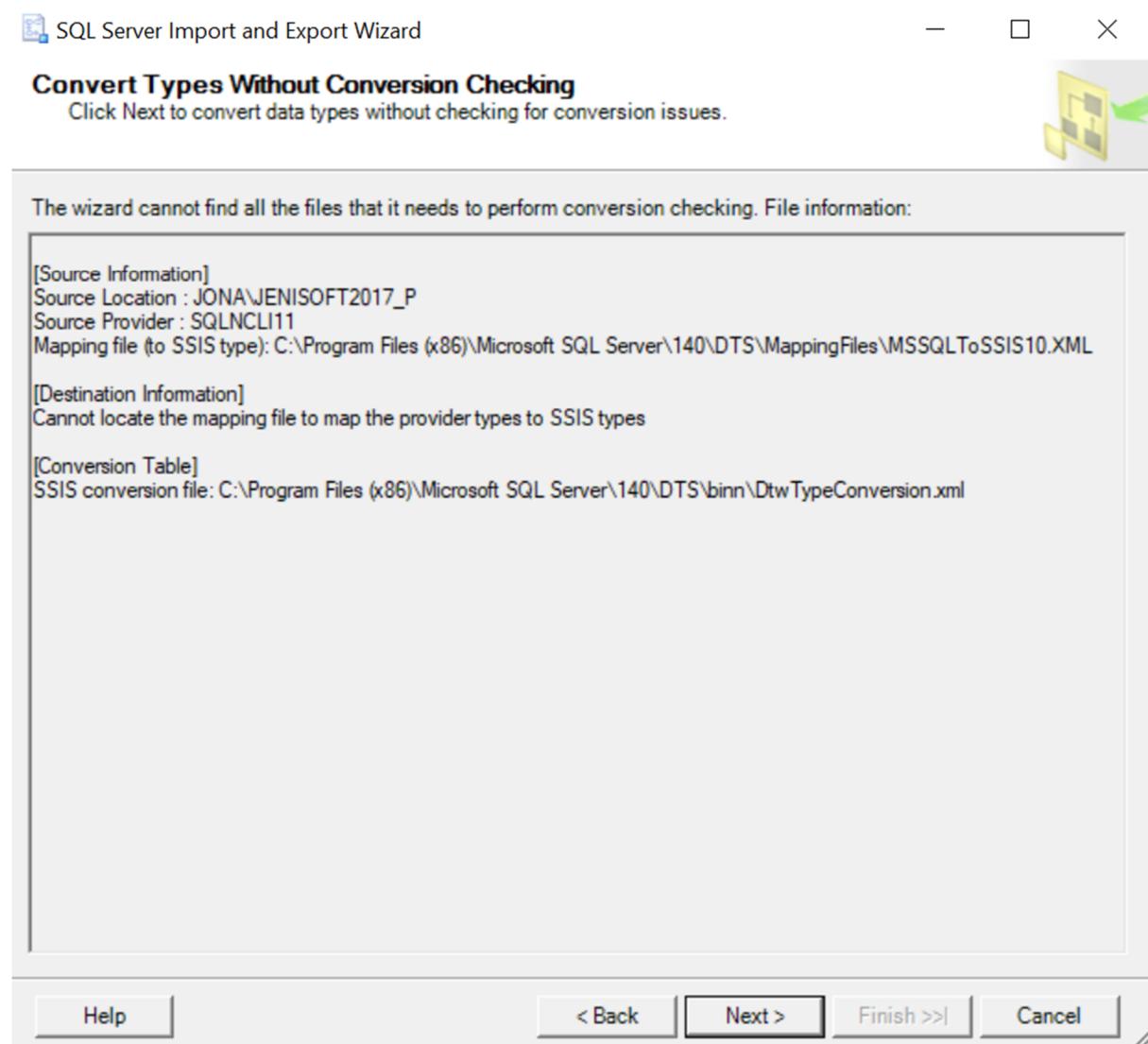


**Here I am migrating only one table like autors table and remaining tables also need to follow the same**

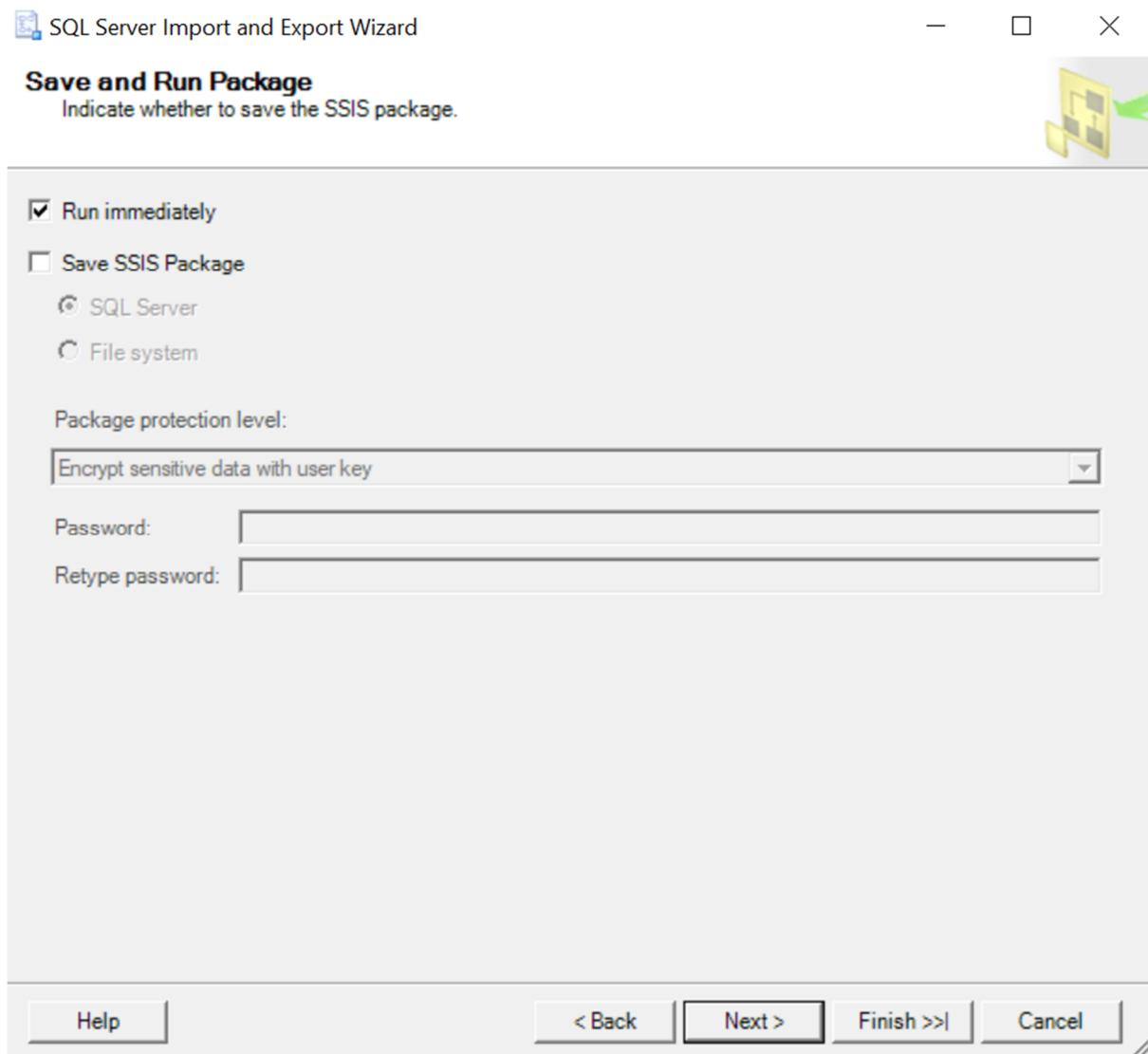
## Click Next



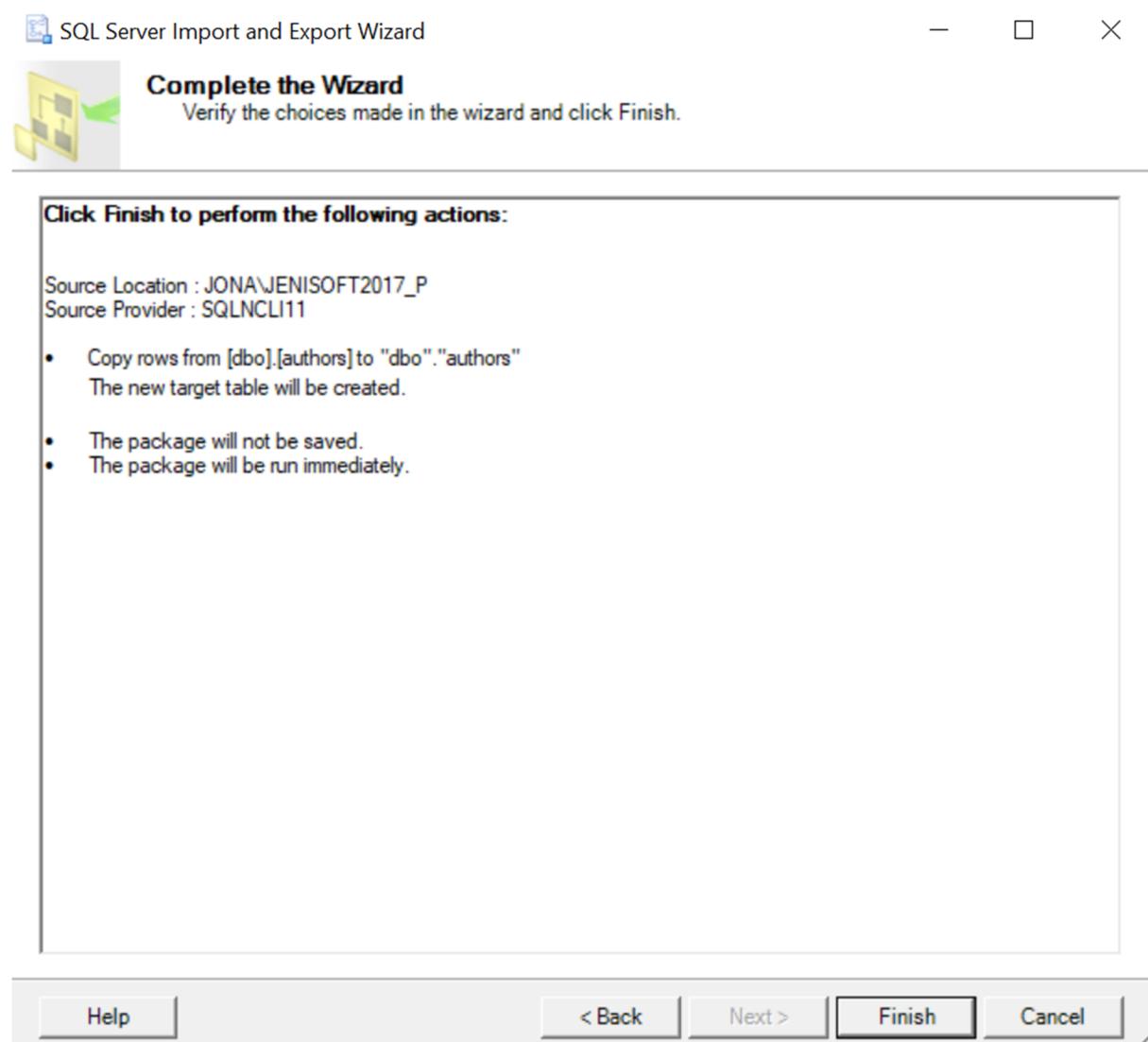
## Click Next



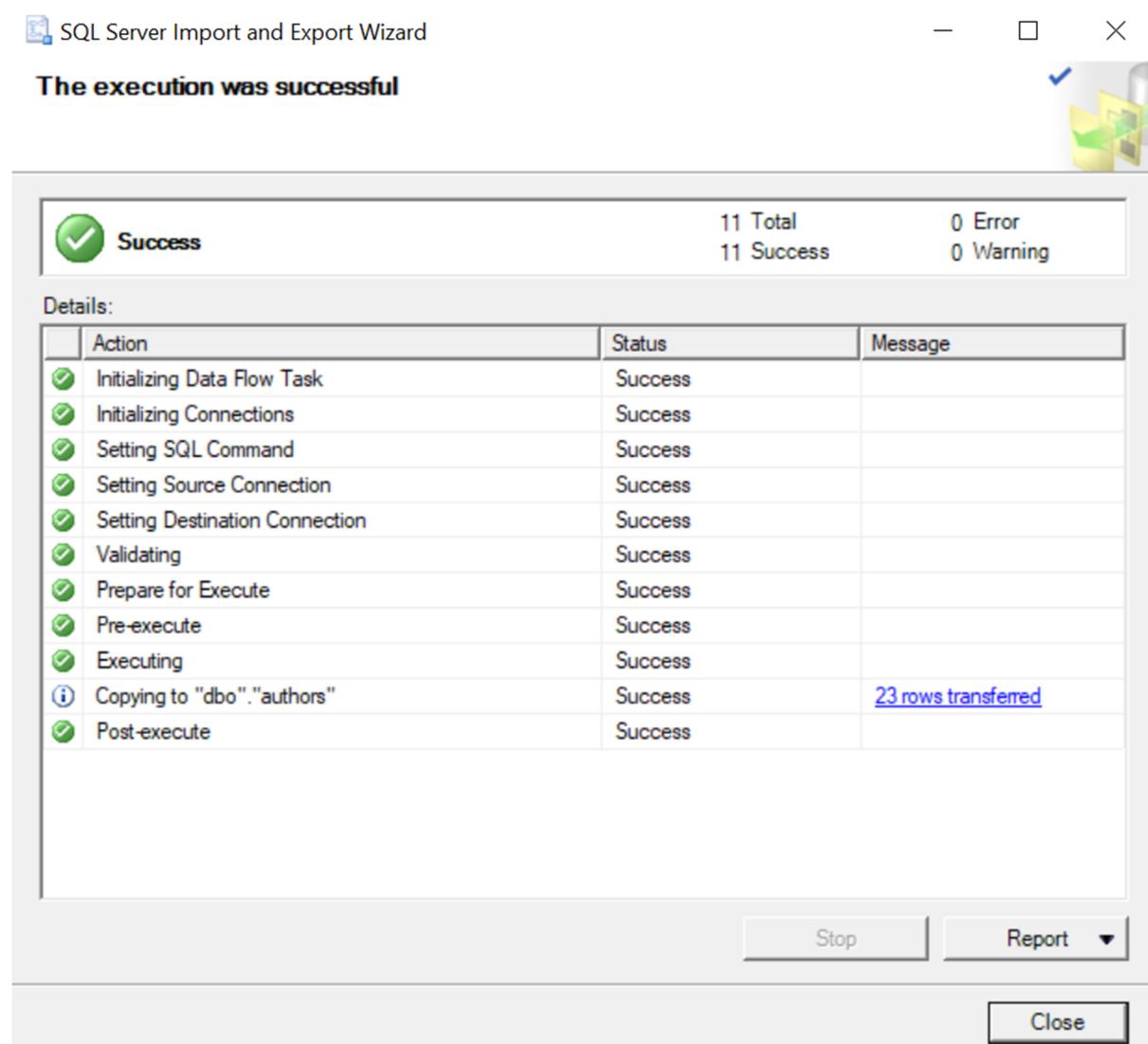
**Click Next**



## Click Finish

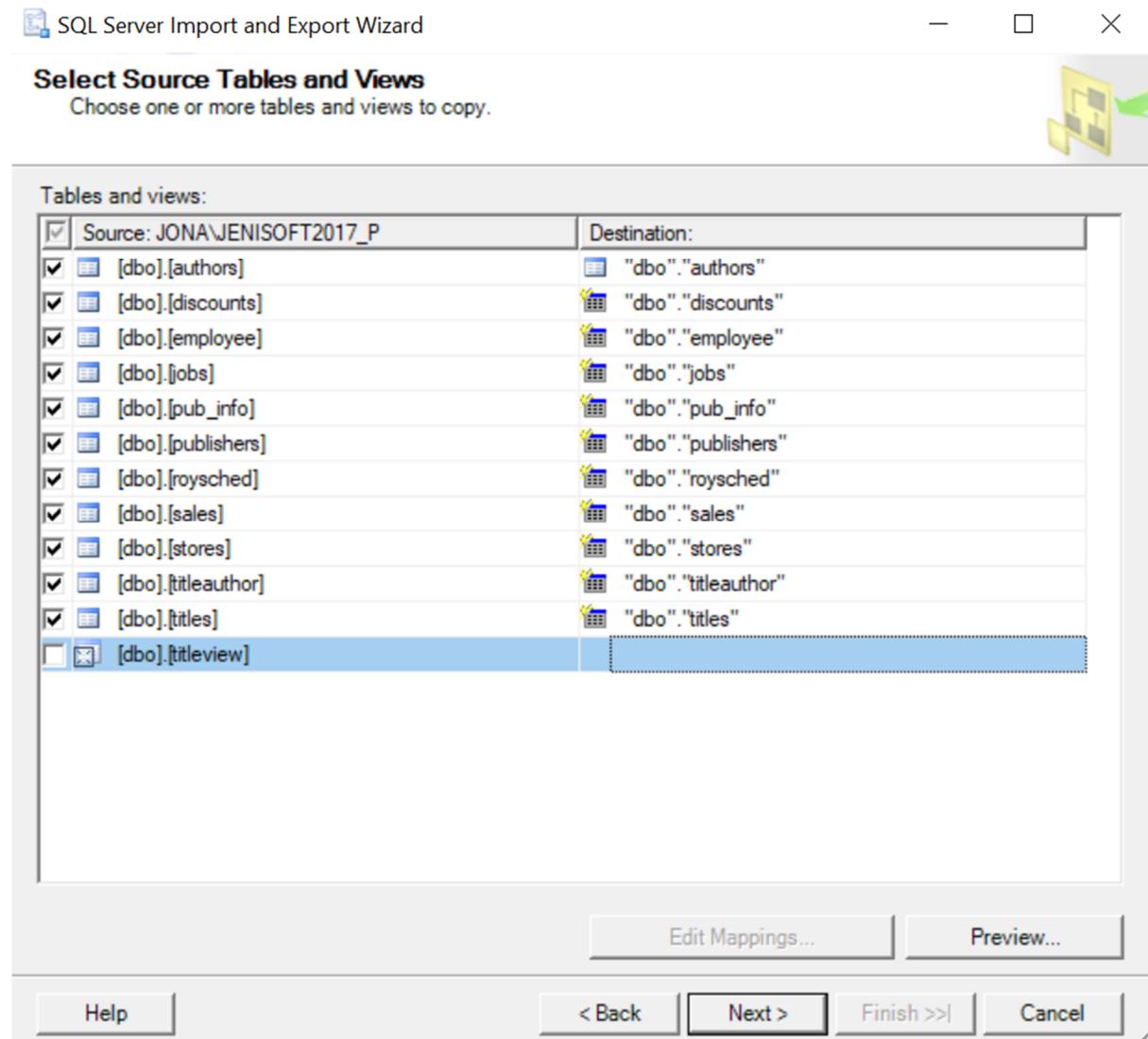


**Click Close**

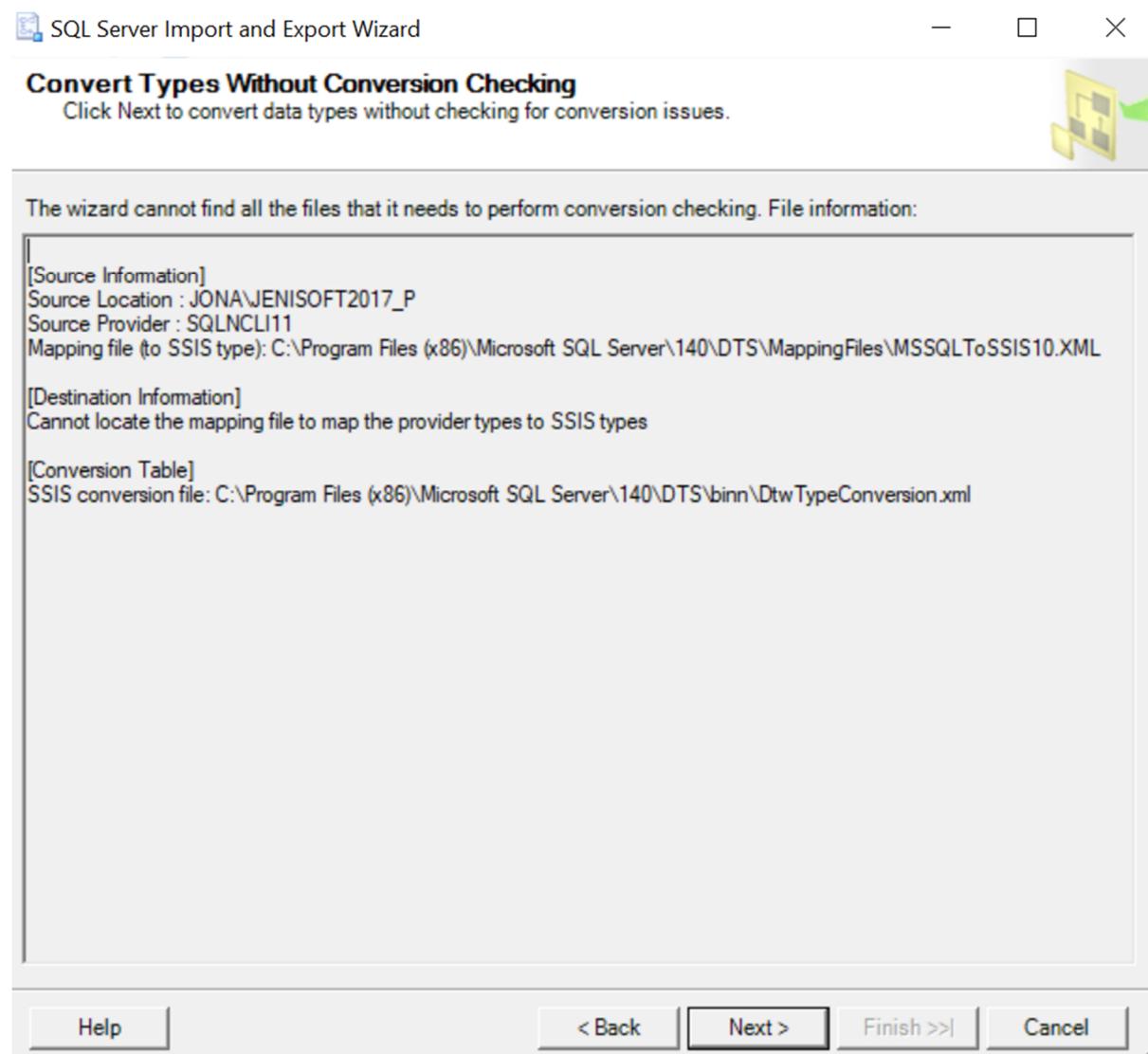


**Now All tables are migrated from MS SQL to PG and Done all the tables source and destination Mappings**

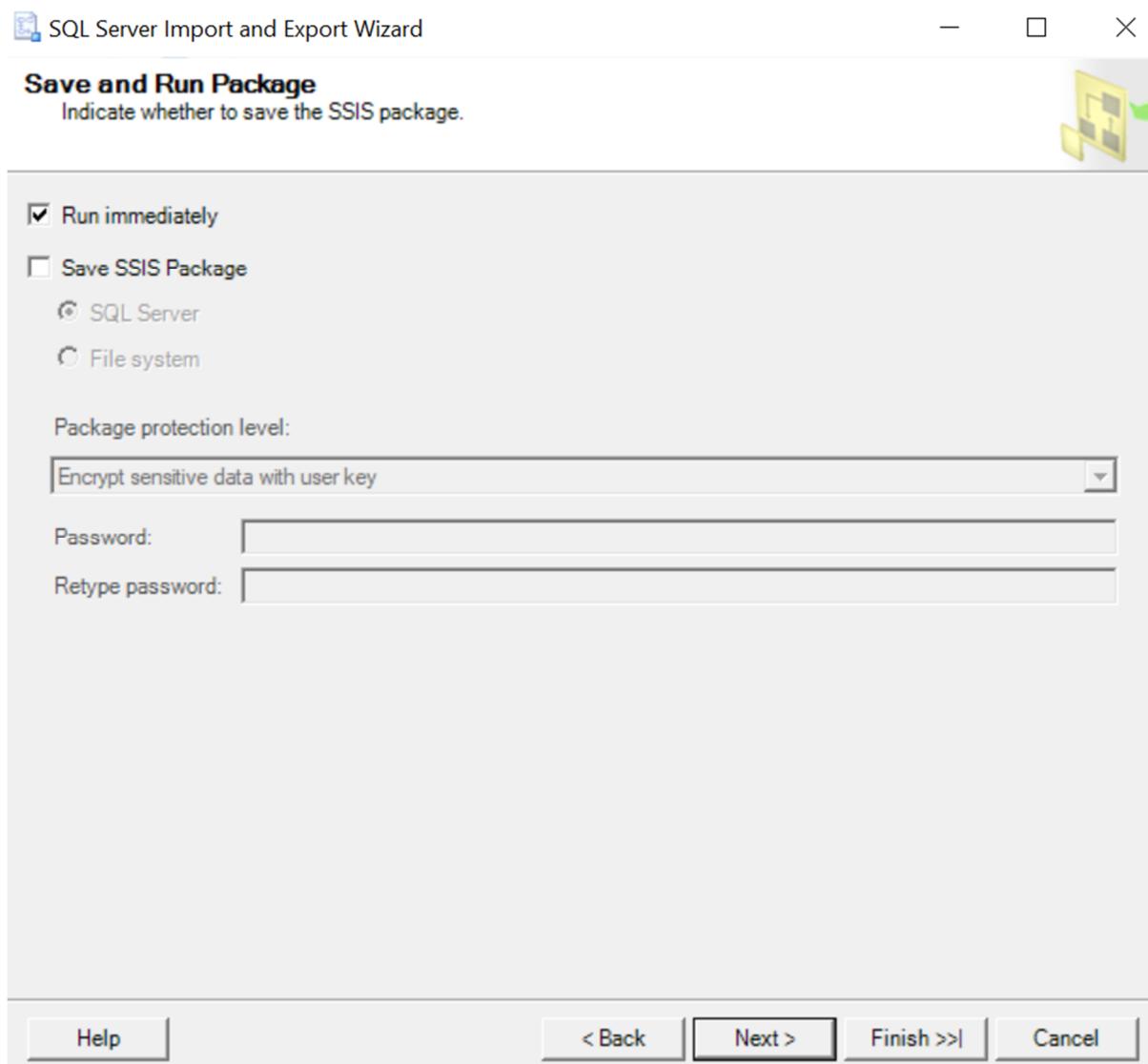
**Click Next**



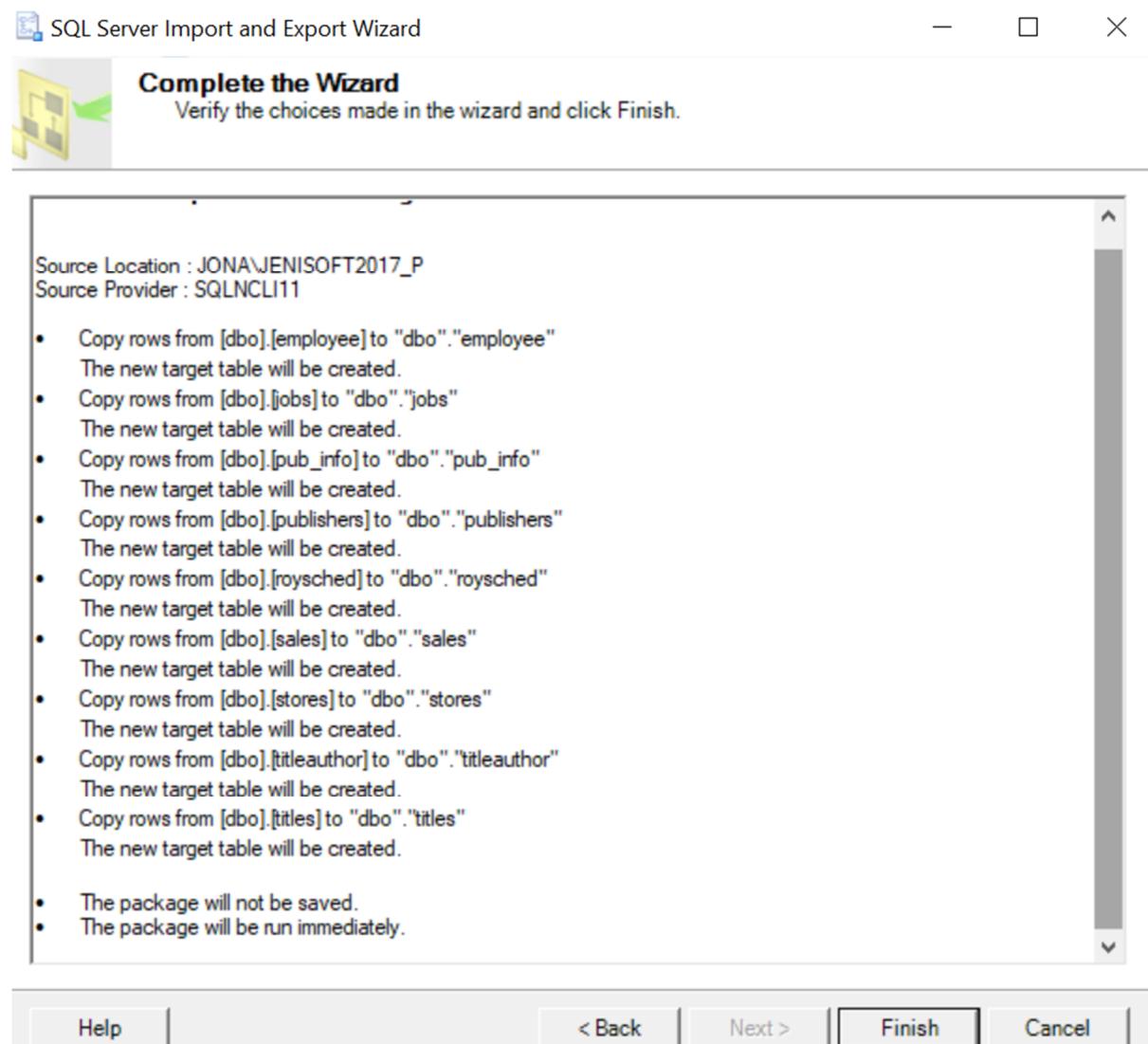
## Click Next



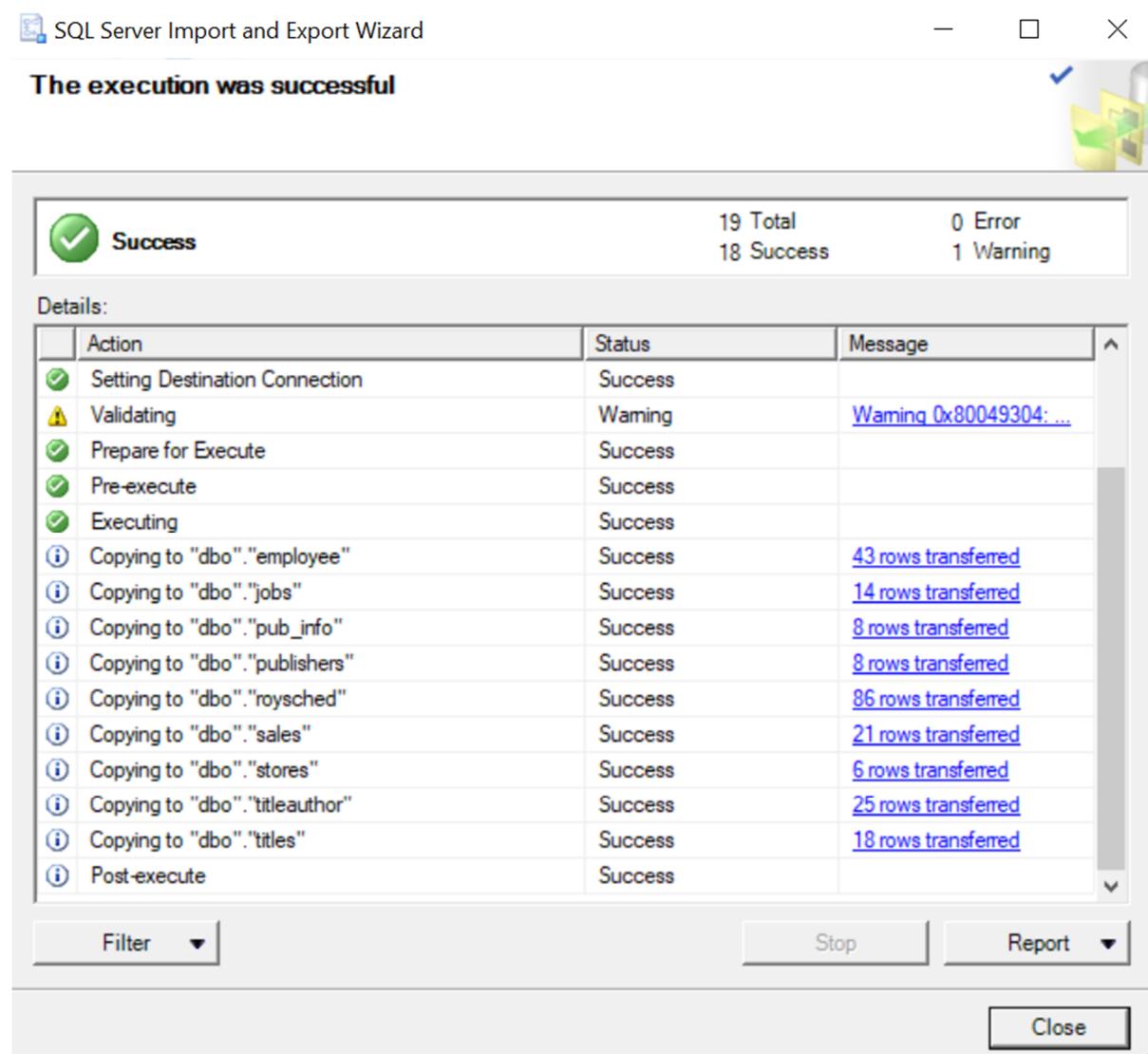
## Click Next



## Click Finish



**Click Close**



All the table and data migration completed.

## Create remaining all object in PG database using SQL commands:

1. Views
2. Add default column values
3. Add constraints
4. Create procedure / Functions

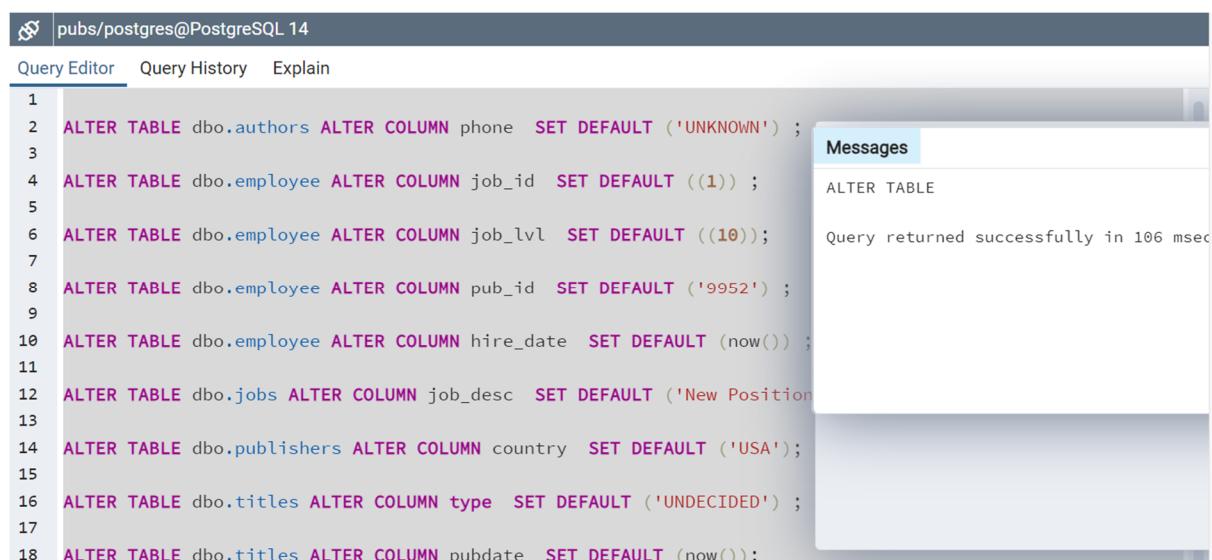
### View:

```
CREATE VIEW dbo.titleview
```

AS

```
select title, au_ord, au_lname, price, ytd_sales, pub_id  
from dbo.authors, dbo.titles, dbo.titleauthor  
where authors.au_id = titleauthor.au_id  
AND titles.title_id = titleauthor.title_id;
```

### Set Default value to table columns:



The screenshot shows a PostgreSQL query editor window with a toolbar at the top. The main area contains a series of ALTER TABLE statements numbered 1 through 18. The statements are as follows:

```
1 ALTER TABLE dbo.authors ALTER COLUMN phone SET DEFAULT ('UNKNOWN') ;  
2 ALTER TABLE dbo.employee ALTER COLUMN job_id SET DEFAULT ((1)) ;  
3 ALTER TABLE dbo.employee ALTER COLUMN job_lvl SET DEFAULT ((10));  
4 ALTER TABLE dbo.employee ALTER COLUMN pub_id SET DEFAULT ('9952') ;  
5 ALTER TABLE dbo.employee ALTER COLUMN hire_date SET DEFAULT (now()) ;  
6 ALTER TABLE dbo.jobs ALTER COLUMN job_desc SET DEFAULT ('New Position');  
7 ALTER TABLE dbo.publishers ALTER COLUMN country SET DEFAULT ('USA');  
8 ALTER TABLE dbo.titles ALTER COLUMN type SET DEFAULT ('UNDECIDED') ;  
9 ALTER TABLE dbo.titles ALTER COLUMN pubdate SET DEFAULT (now());  
10 ALTER TABLE dbo.titles ALTER COLUMN pubdate SET DEFAULT (now());  
11 ALTER TABLE dbo.titles ALTER COLUMN pubdate SET DEFAULT (now());  
12 ALTER TABLE dbo.titles ALTER COLUMN pubdate SET DEFAULT (now());  
13 ALTER TABLE dbo.titles ALTER COLUMN pubdate SET DEFAULT (now());  
14 ALTER TABLE dbo.titles ALTER COLUMN pubdate SET DEFAULT (now());  
15 ALTER TABLE dbo.titles ALTER COLUMN pubdate SET DEFAULT (now());  
16 ALTER TABLE dbo.titles ALTER COLUMN pubdate SET DEFAULT (now());  
17 ALTER TABLE dbo.titles ALTER COLUMN pubdate SET DEFAULT (now());  
18 ALTER TABLE dbo.titles ALTER COLUMN pubdate SET DEFAULT (now());
```

The right side of the window shows a "Messages" panel with the text "ALTER TABLE" and "Query returned successfully in 106 msec".

```
ALTER TABLE dbo.authors ALTER COLUMN phone SET DEFAULT ('UNKNOWN') ;
```

```
ALTER TABLE dbo.employee ALTER COLUMN job_id SET DEFAULT ((1)) ;
```

```
ALTER TABLE dbo.employee ALTER COLUMN job_lvl SET DEFAULT ((10));
```

```
ALTER TABLE dbo.employee ALTER COLUMN pub_id SET DEFAULT ('9952');
```

```
ALTER TABLE dbo.employee ALTER COLUMN hire_date SET DEFAULT (now());
```

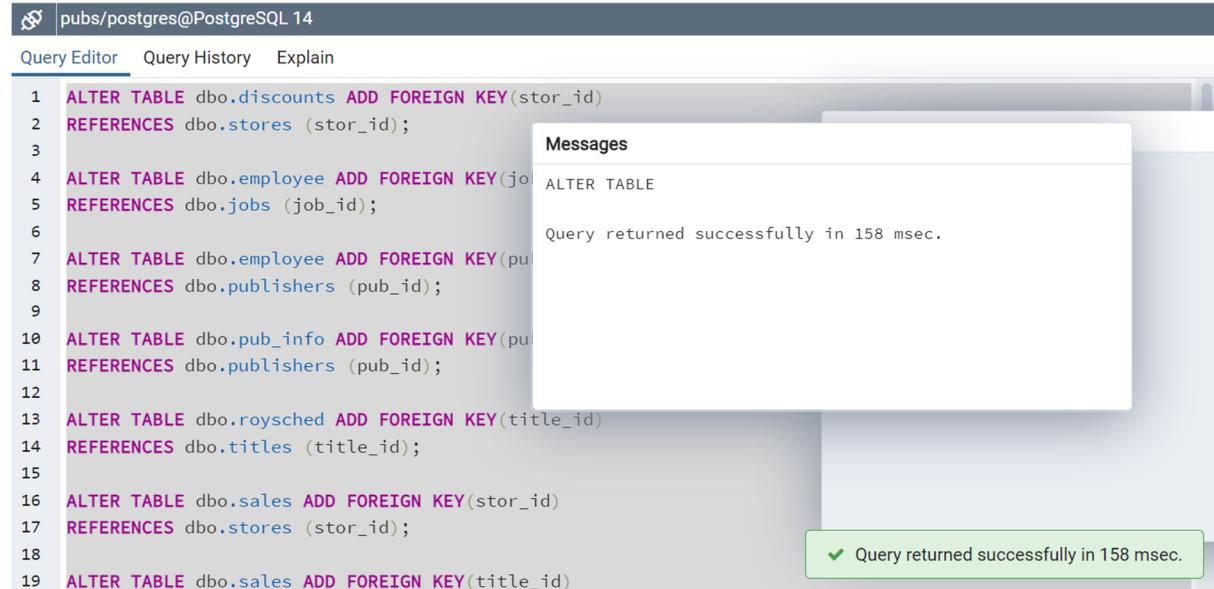
```
ALTER TABLE dbo.jobs ALTER COLUMN job_desc SET DEFAULT ('New Position - title not formalized yet');
```

```
ALTER TABLE dbo.publishers ALTER COLUMN country SET DEFAULT ('USA');
```

```
ALTER TABLE dbo.titles ALTER COLUMN type SET DEFAULT ('UNDECIDED');
```

```
ALTER TABLE dbo.titles ALTER COLUMN pubdate SET DEFAULT (now());
```

### Add Constraints on PG tables:



The screenshot shows a pgAdmin interface with a query editor window. The query editor contains the following SQL code, which consists of 19 numbered lines. Lines 1 through 18 are part of a single transaction, and line 19 is a separate transaction.

```
1 ALTER TABLE dbo.discounts ADD FOREIGN KEY(stor_id)
2 REFERENCES dbo.stores (stor_id);
3
4 ALTER TABLE dbo.employee ADD FOREIGN KEY(job_id)
5 REFERENCES dbo.jobs (job_id);
6
7 ALTER TABLE dbo.employee ADD FOREIGN KEY(pub_id)
8 REFERENCES dbo.publishers (pub_id);
9
10 ALTER TABLE dbo.pub_info ADD FOREIGN KEY(pub_id)
11 REFERENCES dbo.publishers (pub_id);
12
13 ALTER TABLE dbo.roysched ADD FOREIGN KEY(title_id)
14 REFERENCES dbo.titles (title_id);
15
16 ALTER TABLE dbo.sales ADD FOREIGN KEY(stor_id)
17 REFERENCES dbo.stores (stor_id);
18
19 ALTER TABLE dbo.sales ADD FOREIGN KEY(title_id)
```

A message box titled "Messages" appears over the code area, containing the text "ALTER TABLE" and "Query returned successfully in 158 msec." A green success message at the bottom right of the screen also states "Query returned successfully in 158 msec."

```
ALTER TABLE dbo.discounts ADD FOREIGN KEY(stor_id)
```

```
REFERENCES dbo.stores (stor_id);
```

```
ALTER TABLE dbo.employee ADD FOREIGN KEY(job_id)
```

**REFERENCES dbo.jobs (job\_id);**

**ALTER TABLE dbo.employee ADD FOREIGN KEY(pub\_id)**

**REFERENCES dbo.publishers (pub\_id);**

**ALTER TABLE dbo.pub\_info ADD FOREIGN KEY(pub\_id)**

**REFERENCES dbo.publishers (pub\_id);**

**ALTER TABLE dbo.roysched ADD FOREIGN KEY(title\_id)**

**REFERENCES dbo.titles (title\_id);**

**ALTER TABLE dbo.sales ADD FOREIGN KEY(stor\_id)**

**REFERENCES dbo.stores (stor\_id);**

**ALTER TABLE dbo.sales ADD FOREIGN KEY(title\_id)**

**REFERENCES dbo.titles (title\_id);**

**ALTER TABLE dbo.titleauthor ADD FOREIGN KEY(au\_id)**

**REFERENCES dbo.authors (au\_id);**

**ALTER TABLE dbo.titleauthor ADD FOREIGN KEY(title\_id)**

**REFERENCES dbo.titles (title\_id);**

**ALTER TABLE dbo.titles ADD FOREIGN KEY(pub\_id)**

**REFERENCES dbo.publishers (pub\_id);**

```
ALTER TABLE dbo.authors ADD CHECK ((au_id like '[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9][0-9][0-9]'));
```

```
ALTER TABLE dbo.authors ADD CHECK ((zip like '[0-9][0-9][0-9][0-9][0-9]'));
```

```
TER TABLE dbo.jobs ADD CHECK ((max_lvl<=(250)));
```

```
ALTER TABLE dbo.jobs ADD CHECK ((min_lvl>=(10)));
```

### Create procedure on PG

The screenshot shows the pgAdmin interface with a query editor window. The code being run is:

```
40
41
42 CREATE OR REPLACE PROCEDURE dbo.reptq3 ( p_
43 p_type char(12), OUT cur REFCURSOR)
44 AS $$
```

A dropdown menu is open over the code, showing the following options:

- Messages
- CREATE PROCEDURE

The message area displays:

```
Query returned successfully in 147 msec.
```

At the bottom right of the interface, there is a green status bar with the message:

```
✓ Query returned successfully in 147 msec.
```

```
CREATE OR REPLACE PROCEDURE dbo.byroyalty ( p_percentage int, OUT cur REFCURSOR)
```

```
AS $$
```

```
BEGIN
```

```
open cur for select au_id from dbo.titleauthor
```

```
where titleauthor.royaltyper = p_percentage;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE PROCEDURE dbo.reptq1 (OUT cur REFCURSOR) AS $$  
BEGIN  
open cur for select  
    case when grouping(pub_id) = 1 then 'ALL' else pub_id end as pub_id,  
        avg(price) as avg_price  
from dbo.titles  
where price is NOT NULL  
group by rollup (pub_id)  
order by pub_id;  
END;  
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE PROCEDURE dbo.reptq2 (OUT cur REFCURSOR) AS $$  
BEGIN  
open cur for select  
    case when grouping(type) = 1 then 'ALL' else type end as type,  
        case when grouping(pub_id) = 1 then 'ALL' else pub_id end as pub_id,  
            avg(ytd_sales) as avg_ytd_sales  
from dbo.titles  
where pub_id is NOT NULL  
group by rollup (pub_id, type);  
  
END;  
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE PROCEDURE dbo.reptq3 ( p_lolimit money, p_hilimit money,
p_type char(12), OUT cur REFCURSOR)
AS $$

BEGIN

open cur for select

    case when grouping(pub_id) = 1 then 'ALL' else pub_id end as pub_id,
    case when grouping(type) = 1 then 'ALL' else type end as type,
    count(title_id) as cnt

from dbo.titles

where price >p_lolimit AND price <p_hilimit AND type = p_type OR type LIKE '%cook%'

group by rollup (pub_id, type);

END;

$$ LANGUAGE plpgsql;
```

Migration issues:

Check Constraints issues and not able to create the following constraints to PG

```
ALTER TABLE dbo.authors ADD CHECK ((au_id like '[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9][0-9][0-9]'));
```

ERROR: check constraint "authors\_au\_id\_check" of relation "authors" is violated by some row

SQL state: 23514

```
ALTER TABLE dbo.authors ADD CHECK ((zip like '[0-9][0-9][0-9][0-9][0-9]'));
```

ERROR: check constraint "authors\_zip\_check" of relation "authors" is violated by some row

SQL state: 23514

```
ALTER TABLE employee ADD CONSTRAINT CK_emp_id CHECK ((emp_id like '[A-Z][A-Z][A-Z][1-9][0-9][0-9][0-9][0-9][FM]' OR emp_id like '[A-Z]-[A-Z][1-9][0-9][0-9][0-9][0-9][FM]'));
```

ERROR: check constraint "ck\_emp\_id" of relation "employee" is violated by some row

SQL state: 23514

```
ALTER TABLE employee VALIDATE CONSTRAINT CK_emp_id;
```

```
ALTER TABLE publishers ADD CHECK ((pub_id='1756' OR pub_id='1622' OR pub_id='0877' OR pub_id='0736' OR pub_id='1389' OR pub_id like '99[0-9][0-9]'));
```

ERROR: check constraint "publishers\_pub\_id\_check" of relation "publishers" is violated by some row

SQL state: 23514

**Function creation with issues on PG and need developer support to modify the function**

**CREATE OR REPLACE FUNCTION dbo.employee\_insupd\_TRIGGER\_FUNCTION()**

**RETURNS TRIGGER**

**AS \$\$**

**BEGIN**

--Get the range of level for this job type from the jobs table.

**declare v\_min\_lvl smallint;**

**v\_max\_lvl smallint;**

**v\_emp\_lvl smallint;**

**v\_job\_id smallint;**

**select j.min\_lvl,**

**j.max\_lvl,**

**i.job\_lvl,**

**i.job\_id into v\_min\_lvl, v\_max\_lvl, v\_emp\_lvl, v\_job\_id**

**from dbo.employee e, dbo.jobs j, inserted i**

**where e.emp\_id = i.emp\_id AND i.job\_id = j.job\_id;**

**IF (v\_job\_id = 1) and (v\_emp\_lvl <> 10)**

**then**

**raise exception 'Job id 1 expects the default level of 10.'**

**ROLLBACK TRANSACTION;**

**ELSEIF**

**NOT (v\_emp\_lvl BETWEEN v\_min\_lvl AND v\_max\_lvl)**

**then**

**raise exception 'The level for job\_id:%d should be between %d and %d.'**

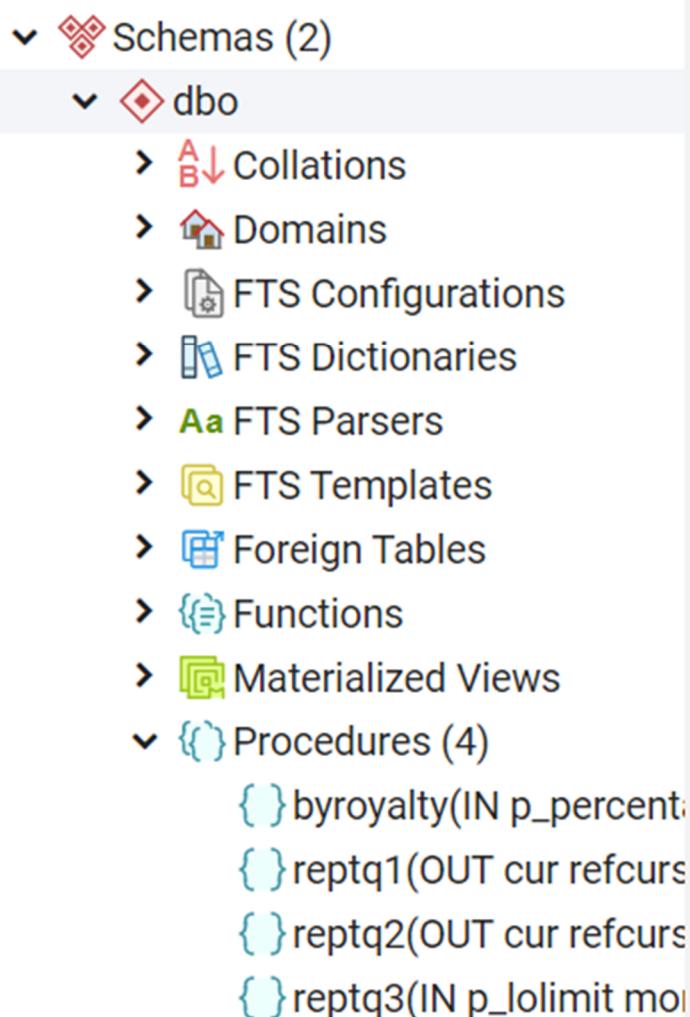
**16, 1, v\_job\_id, v\_min\_lvl, v\_max\_lvl)**

**ROLLBACK TRANSACTION;**

```
END IF;  
END;  
$$ LANGUAGE PLPGSQL;
```

```
CREATE TRIGGER dbo.employee_insupd  
FOR insert OR UPDATE  
ON dbo.employee  
FOR EACH ROW  
EXECUTE PROCEDURE employee_insupd_TRIGGER_FUNCTION();
```

#### Migrated object lists on PG on Procedures



Migrated list of tables:

Browser

- ❖ dbo
  - > Collations
  - > Domains
  - > FTS Configurations
  - > FTS Dictionaries
  - > FTS Parsers
  - > FTS Templates
  - > Foreign Tables
  - > Functions
  - > Materialized Views
  - > Procedures (4)
  - > Sequences
- ❖ Tables (11)
  - > authors
  - > discounts
  - > employee
  - > jobs
  - > pub\_info
  - > publishers
  - > roysched
  - > sales
  - > stores
  - > titleauthor
  - > titles

Migrated list of types and views

- ▼ Schemas (2)
  - ▼ dbo
    - > Collations
    - > Domains
    - > FTS Configurations
    - > FTS Dictionaries
    - > FTS Parsers
    - > FTS Templates
    - > Foreign Tables
    - > Functions
    - > Materialized Views
    - > Procedures (4)
    - > Sequences
    - > Tables (11)
  - > Trigger Functions
- ▼ Types (3)
  - empid
  - id
  - tid
- ▼ Views (1)
  - ▼ titleview
    - > Columns
    - > Rules