# Use AWS DMS to migrate data to AWS or Azure PostgreSQL databases

As outlined in Migration tools, the *AWS Schema Conversion Tool* (SCT), and the *AWS Database Migration Service* (DMS) are tools you will find useful when migrating your database to *either* Cloud used by BNZ (Azure, or AWS).
We used SCT to convert our **database structure** in the convert database objects with SCT example.
The example *Target* database was on Azure, but you can adapt that procedure by choosing an AWS hosted Relational Database System (RDS) instead, such as the one we will be using below.

In this example, we will use DMS to migrate the **data** from our on-premises database, to an AWS RDS.
So, our ***Source*** DB2 LUW database is on an on-premises Windows Server. Our ***Target*** database is an AWS PostgreSQL database.

You can adapt this procedure to migrate your particular database' data to PostgreSQL AWS databases.
The differences are merely the choice of a different *Target* database


**DMS** is used inside your AWS portal. You will need an AWS logon and account to use it.

## Pre-requsites

### IAM Roles

The following IAM roles are required for DMS to function when you launch DMS Service Catalog item.

- dms-vpc-role

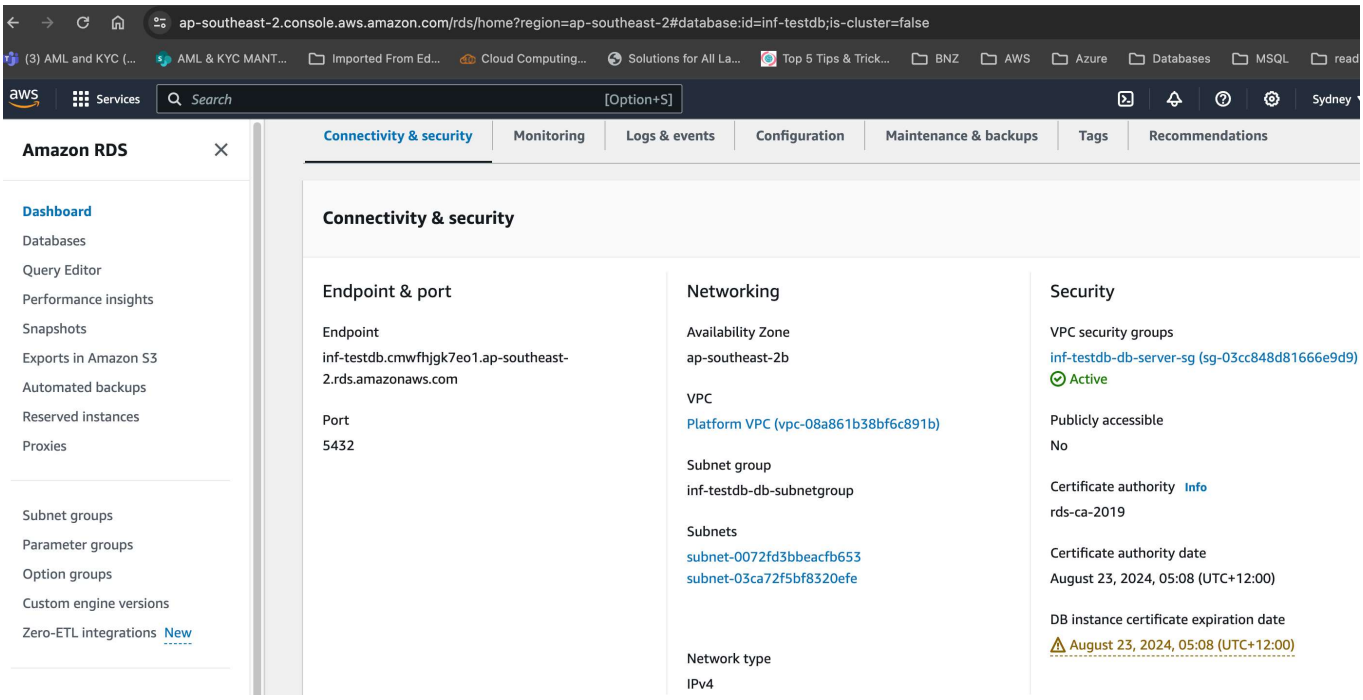
    - Service: dms.ap-southeast-2.amazonaws.com
    - Permissions:
        - BNZ-Account-Policy
        - BNZ-CMK-Usage-RDS
    - Permissions boundary
        - BNZ-Account-Policy
- dms-cloudwatch-logs-role
    - Service: dms.ap-southeast-2.amazonaws.com
    - Permissions:
        - BNZ-Account-Policy
        - BNZ-CMK-Usage-RDS
    - Permissions boundary
        - BNZ-Account-Policy

You may create these via AWS Console or your approved pipelines.

These IAM roles are to be created ONLY ONCE, no matter number of DMS provisioned products that you are intent to launch.

# Step 1. Ensure you have already set up your environment and accesses

- Our example *target* database is our test AWS PostgreSQL database, inf-testdb




# Step 2. The procedure create the DMS instance via AWS Service Catalog

The following table describes the various configuration options available to DMS consumers

## Service Catalog item configurations

| Configuration Option | Allowed Values | Default Value | Description |
|---|---|---|---|
| `Provisioned product name` | Alphanumeric string, space is not allowed | Auto generated | Name of the database migration service instance |
| `Product version` | Select the latest version | | The version number the latest version |

## DMS Instance configurations

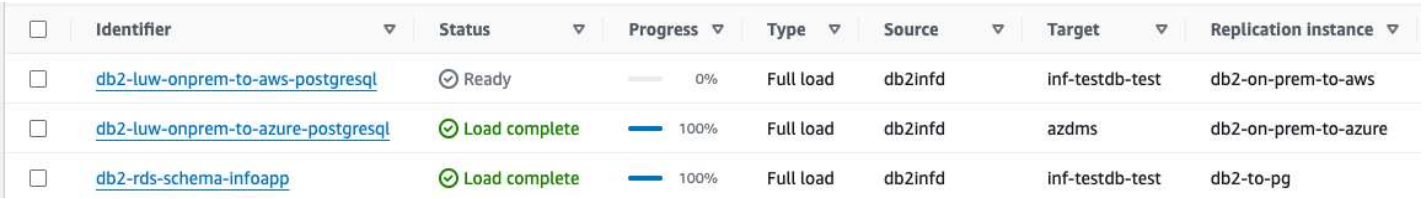
| Configuration Option | Allowed Values | Default Value | Description |
|---|---|---|---|
| `Replication Name` | Alphanumeric string, space is not allowed | | Name of the database migration service instance |
| `Instance Class` | Micro, Small, Medium, Large & XLarge | | Size of the replication instance, recommended Small or Medium unless you have significant content in your source database to migrate. |
| `Instance Class` | Micro, Small, Medium, Large & XLarge | | Size of the replication instance, recommended Small or Medium unless you have significant content in your source database to migrate. |
| `High Availability` | false or true | | Unless you have a requirement for ongoing replication or long running replication, you may not need high availability here. |

## DMS Target Datasource configurations

| Configuration Option | Allowed Values | Default Value | Description |
|---|---|---|---|
| `Target Database` | Alphanumeric string, space is not allowed | | Please provide the database name for the target database. |
| `Target Engine` | DB2, Aurora, Aurora-PostgreSQL, DynamoDB, MySQL, Oracle, PostgreSQL and SQLServer | | Please provide the database engine for the target database. |

| Configuration Option | Allowed Values | Default Value | Description |
|---|---|---|---|
| `Target Host` | BNZ's Private IP address, must be within RFC1918 range. | | Please provide the database IP address for the target database. (Please note this must be an IP address, not the DNS) |
| `Target Port` | Valid target database port | | Please provide the database port for the target database. |
| `Source Schema` | Valid schema name in the target database | | Please provide the database name for the target database schema. |
| `Target Username` | Valid username in the target database | | Please provide the database username for the target database. |
| `Target Password` | Valid password in the target database | | Please provide the database password for the target database. |

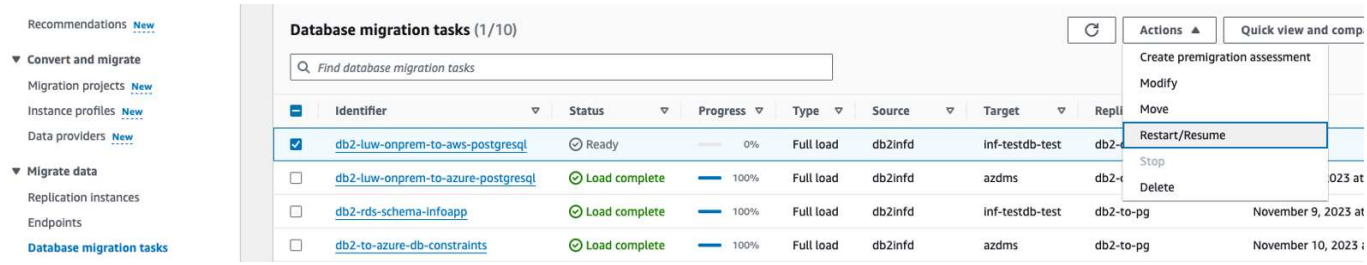
# Step 3. Run migration task



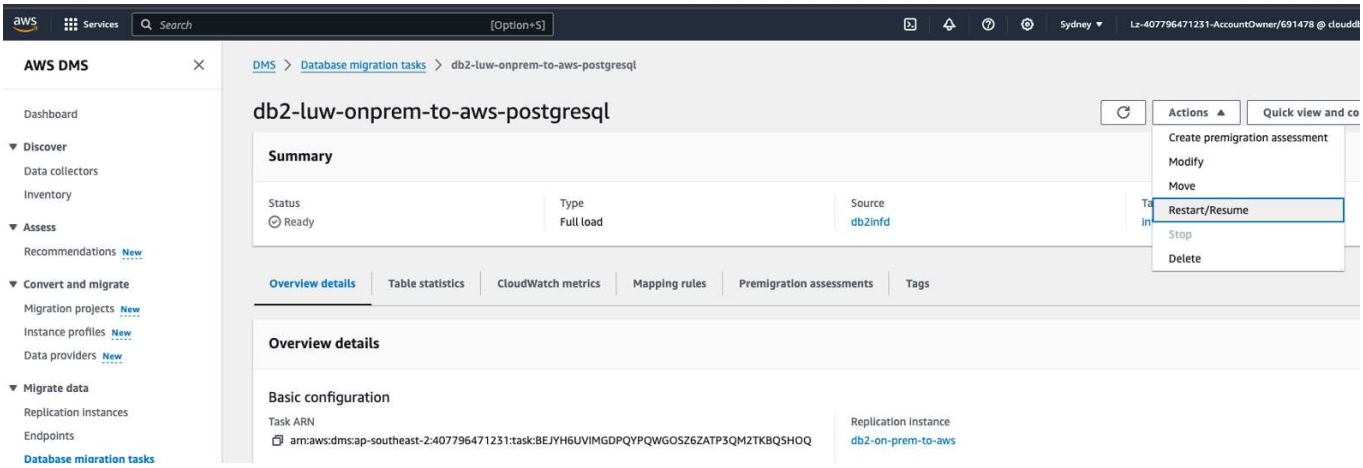
| Identifier | Status | Progress | Type | Source | Target | Replication instance |
|---|---|---|---|---|---|---|
| db2-luw-onprem-to-aws-postgresql | Ready | 0% | Full load | db2infd | inf-testdb-test | db2-on-prem-to-aws |
| db2-luw-onprem-to-azure-postgresql | Load complete | 100% | Full load | db2infd | azdms | db2-on-prem-to-azure |
| db2-rds-schema-infoapp | Load complete | 100% | Full load | db2infd | inf-testdb-test | db2-to-pg |

- Now run the task. You can tick it and run it from the database migration tasks list, or from within your Database migration task. In both cases, you click on the `Actions` button to expand it, then click on `Restart/Resume` to run it.

  Method 1: run from the Database migration tasks list




Method 2: run from inside the open Database migration task



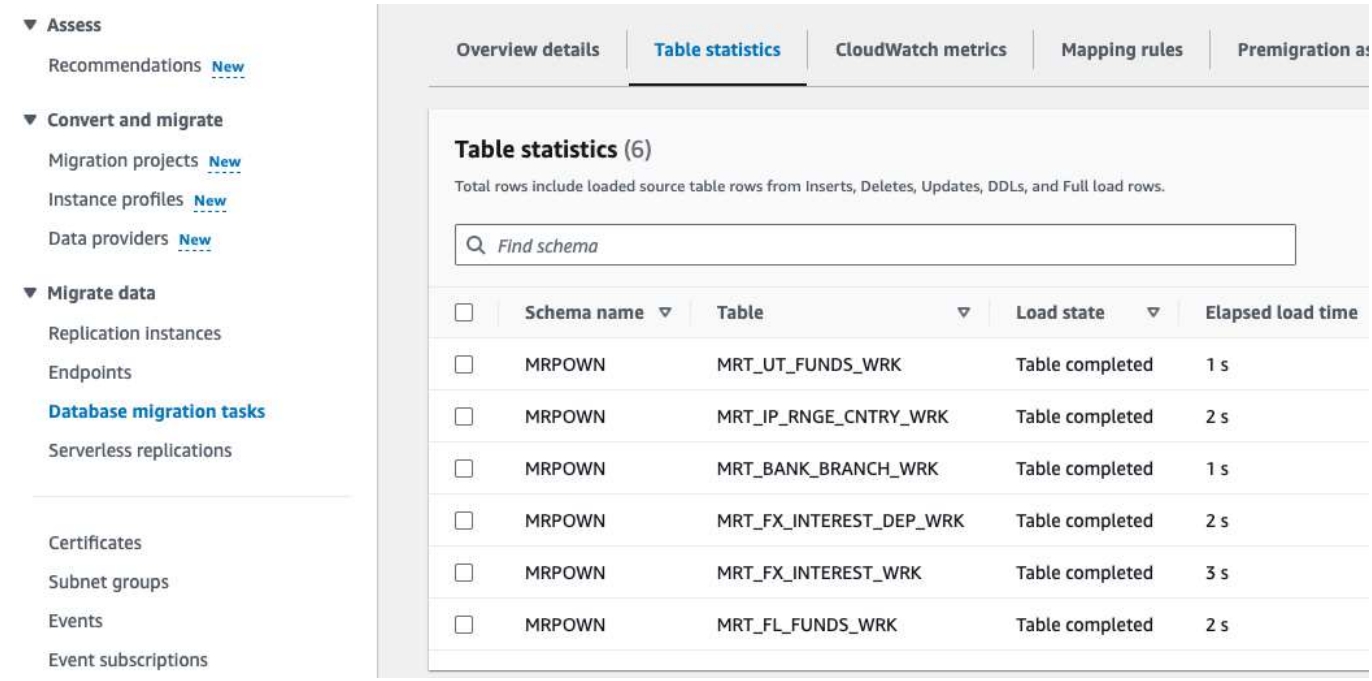

Database migration task (DB2 to AWS PostgreSQL) is starting now:


...and now running:


..and now load complete:


### Check the run statistics are as expected:

- Click on the `Table statistics` tab to see what data was transferred. Scroll to the right to see the rows loaded, otherwise you see a whole lot of zeroes and wonder if anything actually happened.




After scrolling to the right, here are the rows of data loaded:

| Full load rows | Total rows |
|---|---|
| 0 | 0 |
| 0 | 0 |
| 0 | 0 |
| 54 | 54 |
| 8 | 8 |
| 11 | 11 |

- Remember to click on `Create task`, to save your work

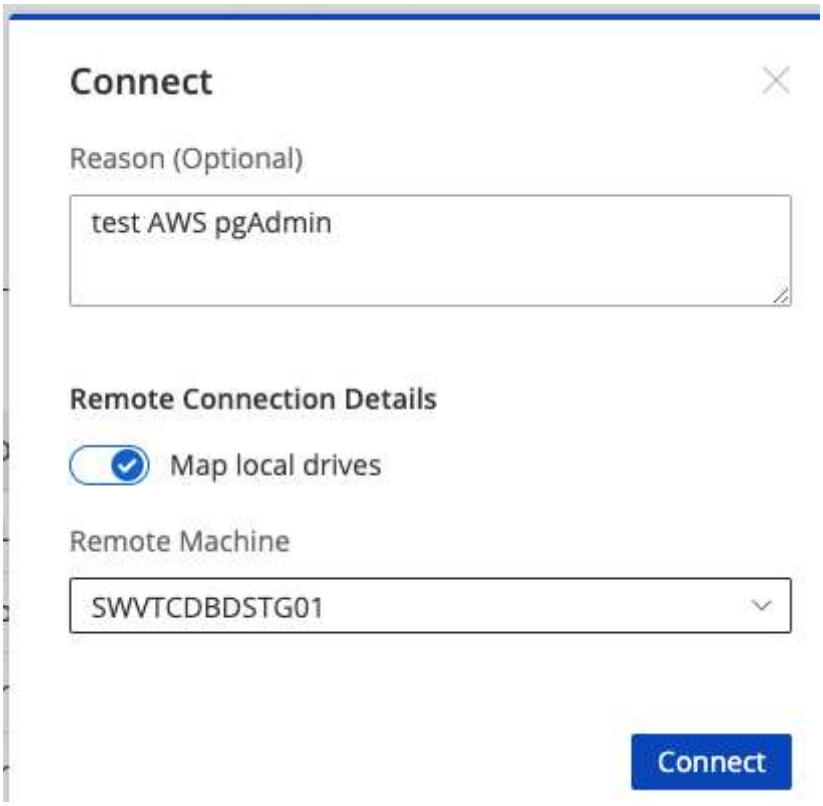

# Step 5. Check the data on your *Target* database

You can carry out any further work on your *Target* database. Or, if you find errors or omissions following the migration, you can delete all or part of the migrated data, and modify your migration process. For

example, you can go back to your **DMS** Migration task and add more *transformation rules*. You may even want to go back to **SCT** (Schema Conversion Tool) again to carry out further changes to the Schema and tables.

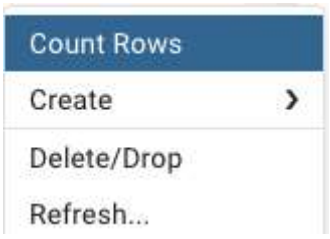
- To administer our AWS PostgreSQL Db, we use pgadmin on the on-premises Windows Server, SWVTCDBDSTG01
- Open your admin Server from CyberArk, and then open pgAdmin (or your particular administration tool).

  RDS Connection:



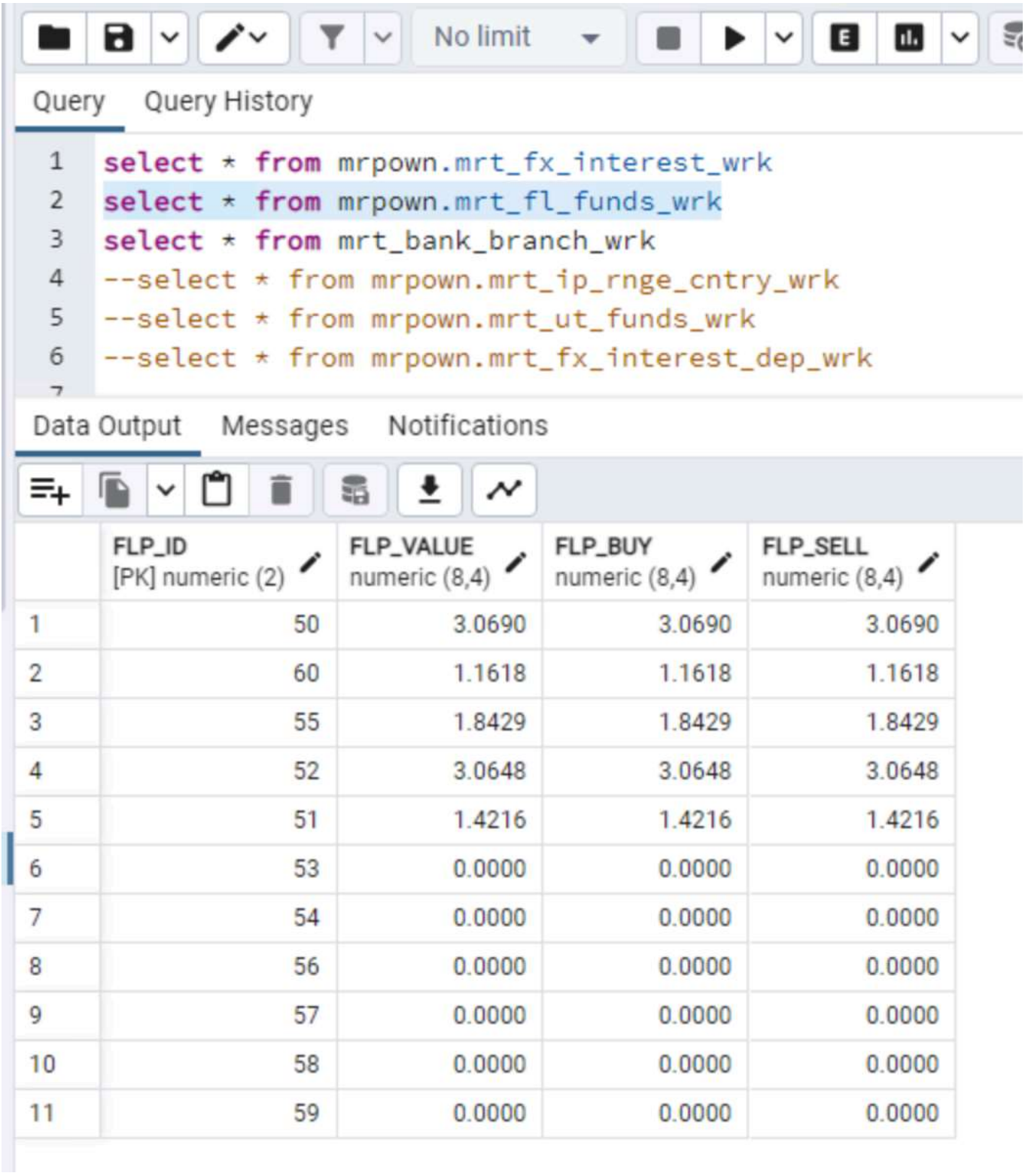

- When the Remote Desktop opens, start pgAdmin from the Windows Start Menu
- Open your Database on the left Menu
- Check the data has come across

  For a quick check, in the left hand menu, drill into your database, open your schema, select a table, right-click, select `Count Rows`.




For more comprehensive checks you can run various SQL queries against your tables:

- In your admin tool, run your SQL to check the database has received the correct data




| | FLP_ID [PK] numeric (2) | FLP_VALUE numeric (8,4) | FLP_BUY numeric (8,4) | FLP_SELL numeric (8,4) |
|---|---|---|---|---|
| 1 | 50 | 3.0690 | 3.0690 | 3.0690 |
| 2 | 60 | 1.1618 | 1.1618 | 1.1618 |
| 3 | 55 | 1.8429 | 1.8429 | 1.8429 |
| 4 | 52 | 3.0648 | 3.0648 | 3.0648 |
| 5 | 51 | 1.4216 | 1.4216 | 1.4216 |
| 6 | 53 | 0.0000 | 0.0000 | 0.0000 |
| 7 | 54 | 0.0000 | 0.0000 | 0.0000 |
| 8 | 56 | 0.0000 | 0.0000 | 0.0000 |
| 9 | 57 | 0.0000 | 0.0000 | 0.0000 |
| 10 | 58 | 0.0000 | 0.0000 | 0.0000 |
| 11 | 59 | 0.0000 | 0.0000 | 0.0000 |

Edit this page