# Download data from Postgres to JSON file and Load them to Snowflake

To copy data from a PostgreSQL table into a JSON format, you can use the PostgreSQL's built-in functions for JSON output combined with the command-line utility `psql`.
Snowflake is extremely performant

## Pre Requisite:

- Heroku CLI is installed on your laptop. Instruction here.
- You can run the command **Heroku PG:PSQL -a dynamartapp-prd**. If not, ensured that Postgres is set up locally. Instruction here.
- Snowsql is installed and configured. Instruction here. Below example for the ~\.snowsql\config file.

```
[connections.prod_bussys]
accountname = wmbjbcq-dt_businesssystems_dw
username = "THIERRY.KRUMEICH@DYNATRACE.COM"
role = BUSSYS_DEVELOPER
dbname = SANDBOX
schema = UPLOAD
authenticator = externalbrowser
```

## Guide

1. Start Window PowerShell (enter PowerShell in the search bar).
2. Create a folder to store PG data if it doesn't exist.

```
md postgres_data
```

3. or navigate to your project directory.

```
Set-Location -Path "c:\users\thierry.krumeich\postgres_data"
```

4. Login to Heroku.

```
heroku login
```

5. Test your query.

```
heroku pg:psql DATABASE_URL --app dynamartapp-prd -c "select id, created_at, updated_at from src_zendesk.ticke
```

6. Export result to JSON and save output to a file.

```
heroku pg:psql DATABASE_URL --app dynamartapp-prd -c "\COPY (SELECT row_to_json(t) FROM (select id, created_at
```

7. Load JSON file to SANDBOX.UPLOAD.JSON_STAGE staging.

```
snowsql -c prod_bussys -q "PUT file://zendesk_ticket.json @json_stage;"
```

8. In Snowsight or using snowsql, Create a temporary table.

```
snowsql -c prod_bussys
```

```
CREATE TEMPORARY TABLE sandbox.upload.zendesk_ticket_json_temp AS SELECT $1::VARIANT as json_data FROM @json_s
```

9. Test the data stored in the temporary table, limited to 10 rowa.

```
SELECT * FROM zendesk_ticket_json_temp LIMIT 10;
```

10. Test the conversion to structure data, limited to 10 rows

```
SELECT $1:"id"::INTEGER as ID, $1:"created_at"::TIMESTAMP as CREATED_AT, $1:"updated_at"::TIMESTAMP as UPDATED
```

11. Create the final table from the temporary table data

```
CREATE TABLE zendesk_ticket AS SELECT $1:"id"::INTEGER AS ID, $1:"created_at"::TIMESTAMP AS CREATED_AT, $1:"up
```

12. Test the final table result, limited to 10 rows.

```
SELECT * FROM zendesk_ticket LIMIT 10;
```

If Snowflake Staging doesn’t exist in a schema, reach out to Snowflake Admin to create the requested staging:

```
$stageName = "PG2SNOW_JSON"
$createQuery = "CREATE STAGE IF NOT EXISTS $stageName;"
$command = "snowsql -c prod_bussys -q `"$createQuery`""
Invoke-Expression -Command $command
```