1. Run the following commands. This is to setup the local repositories to contain information above repository having apache superset

The instruction shown below is taken from here: https://superset.apache.org/docs/installation/pypi

sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt update
sudo apt install python3.11 python3.11-dev python3.11-venv build-essential libssl-dev libffi-dev libsasl2-dev libldap2-dev default-libmysqlclient-dev

```
ardent@ardent:~$ sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt update
sudo apt install python3.11 python3.11-dev python3.11-venv build-essential libss
l-dev libffi-dev libsasl2-dev libldap2-dev default-libmysqlclient-dev
[sudo] password for ardent:
Repository: 'Types: deb
URIs: https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu/
Suites: noble
Components: main
```

Enter your password
Press ENTER when prompted
Then Press y

2. Create new directory superset in the workspace directory and cd into it

```
Processing triggers for libc-bin (2.39-0ubuntu8.4) ...
ardent@ardent:~$ cd Workspace/
ardent@ardent:~/Workspace$ ls
etl  pyspark
ardent@ardent:~/Workspace$ mkdir superset && cd superset
ardent@ardent:~/Workspace/superset$
```

3. Create new virtual environment and then activate it. Make sure you use python 3.11 as 3.12 is not supported by apache superset

```
ardent@ardent:~/Workspace$ mkdir superset && cd superset
ardent@ardent:~/Workspace/superset$ python3.11 -m venv venv
ardent@ardent:~/Workspace/superset$ source venv/bin/activate
(venv) ardent@ardent:~/Workspace/superset$ pip install apache_superset
Collecting apache_superset
```

4. Let us modify the .bashrc as we did for java home and add two parameters. These are required by apache superset. Notice here the difference from JAVA_HOME for java home we just set the variable but here we are exporting it

```
[notice] A new release of pip is available: 24.0 -> 25.1.1
[notice] To update, run: pip install --upgrade pip
(venv) ardent@ardent:~/Workspace/superset$ vim ~/.bashrc
(venv) ardent@ardent:~/Workspace/superset$
```

```
JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
export SUPERSET_SECRET_KEY=YOUR-SECRET-KEY
export FLASK_APP=superset
```

Save and close the .bashrc after adding the SUPERSET_SECRET_KEY and FLASK_APP variables

Then either restart the terminal or source it to update the environment variable. Then we will install Pillow and marshmallow version 3.26.1 using pip

Do not miss these steps as superset requires the specific version and environment variables

```
(venv) ardent@ardent:~/Workspace/superset$ vim ~/.bashrc
(venv) ardent@ardent:~/Workspace/superset$ source ~/.bashrc
ardent@ardent:~/Workspace/superset$ source venv/bin/activate
(venv) ardent@ardent:~/Workspace/superset$ pip install Pillow
Requirement already satisfied: Pillow in ./venv/lib/python3.11/site-packages (11
.2.1)

[notice] A new release of pip is available: 24.0 -> 25.1.1
[notice] To update, run: pip install --upgrade pip
(venv) ardent@ardent:~/Workspace/superset$ pip install marshmallow==3.26.1
Collecting marshmallow==3.26.1
```

5. Initialize the database for superset

```
[notice] A new release of pip is available: 24.0 -> 25.1.1
[notice] To update, run: pip install --upgrade pip
(venv) ardent@ardent:~/Workspace/superset$ superset db upgrade
WARNI [alembic.env] SQLite Database support for metadata databases will
be removed in a future version of Superset.
INFO  [alembic.env] Starting the migration scripts.
INFO  [alembic.runtime.migration] Context impl SQLiteImpl.
INFO  [alembic.runtime.migration] Will assume transactional DDL.
INFO  [alembic.runtime.migration] Running upgrade  -> 4e6a06bad7a8, Init
INFO  [alembic.runtime.migration] Running upgrade 4e6a06bad7a8 -> 5a7bad26f2a7,
empty message
```

6. Let us create a admin user in superset. Enter name, email, and password as per you liking but make sure you remember them

```
INFO  [alembic.env] Migration scripts completed. Duration: 00:00:06
(venv) ardent@ardent:~/Workspace/superset$ superset fab create-admin
Username [admin]:
User first name [admin]:
User last name [user]: admin
Email [admin@fab.org]: admin@admin.com
Password:
Repeat for confirmation:
Recognized Database Authentications.
Admin User admin created.
(venv) ardent@ardent:~/Workspace/superset$
```

7. Load default examples provided by superset

```
Recognized Database Authentications.
Admin User admin created.
(venv) ardent@ardent:~/Workspace/superset$ superset load_examples
2025-06-30 11:31:16,047:INFO:superset.utils.database:Creating database referer
 for examples
2025-06-30 11:31:16,052:INFO:superset.cli.examples:Loading examples metadata a
 related data into examples
```

8. Create default roles and permissions

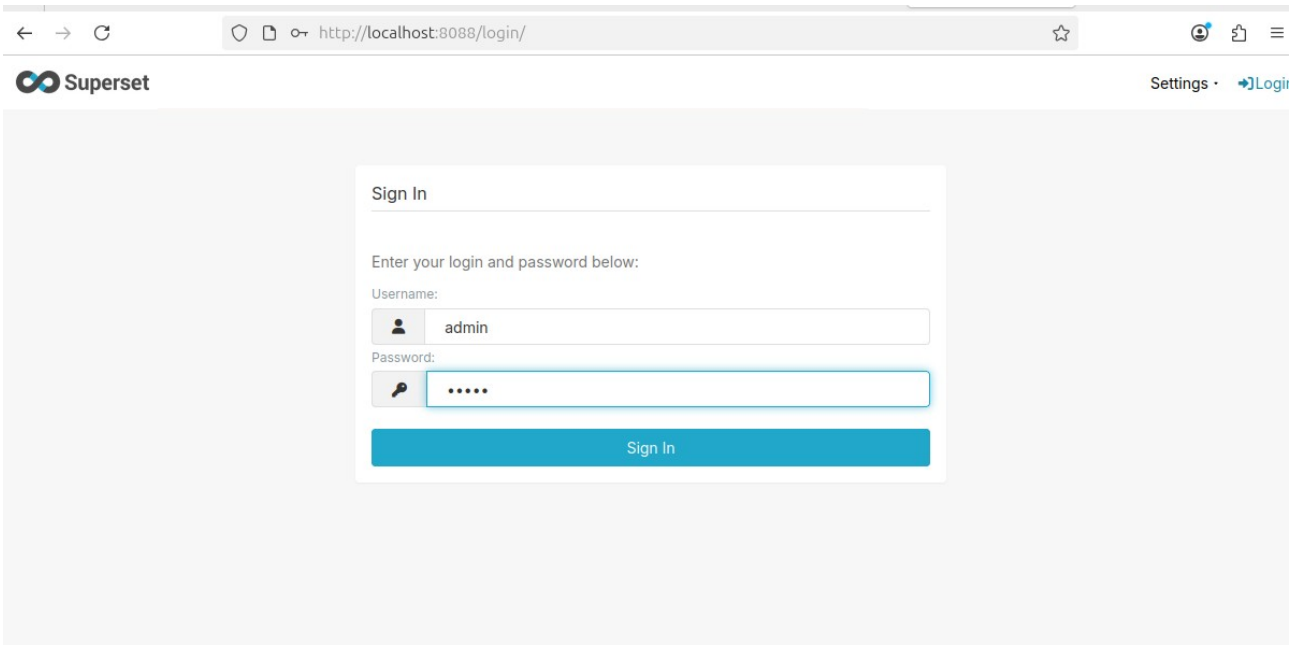
```
master/datasets/examples/slack/channels.csv
2025-06-30 11:32:52,173:WARNING:superset.commands.dataset.importers.v1.utils:L
ding data outside the import transaction
(venv) ardent@ardent:~/Workspace/superset$ superset init
```

The steps before this are one time thing. From now on you just need to activate the virtual environment with superset installed and run the following command to run superset server

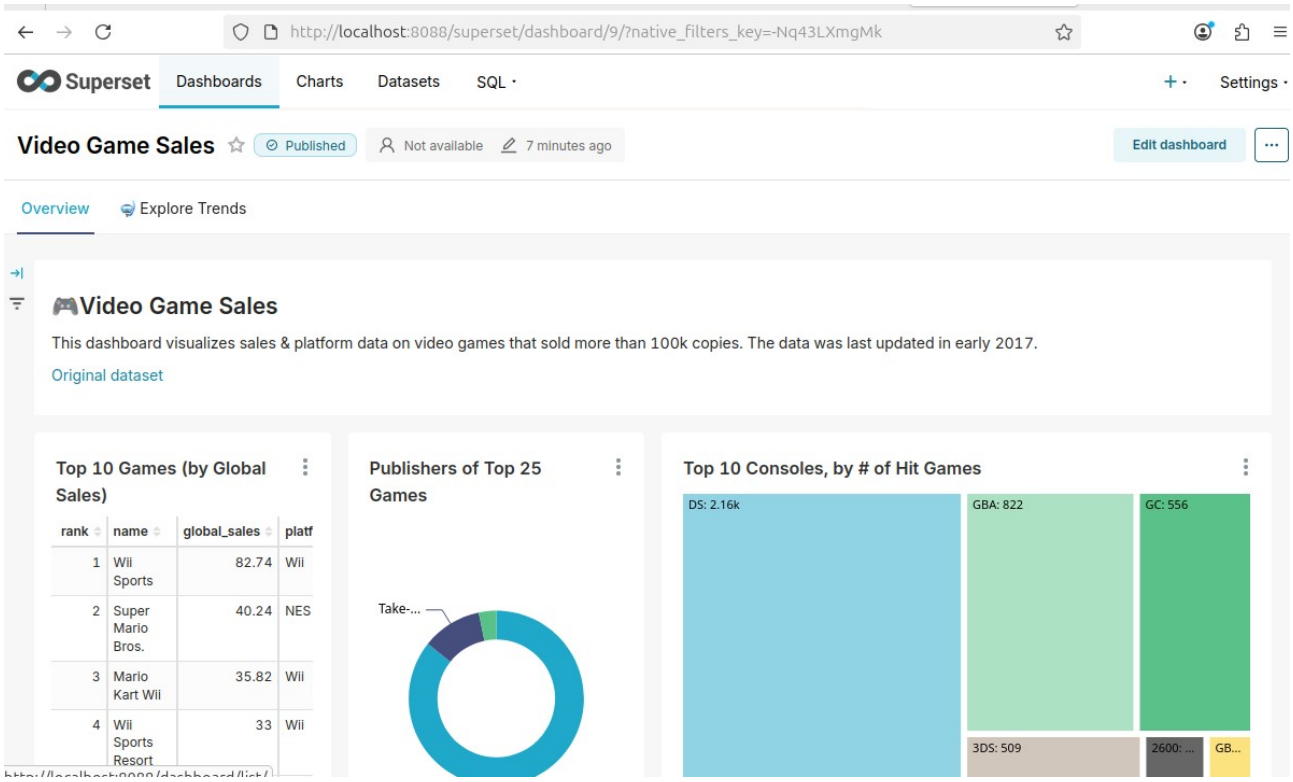
9. Start the superset server in development mode in port 8088

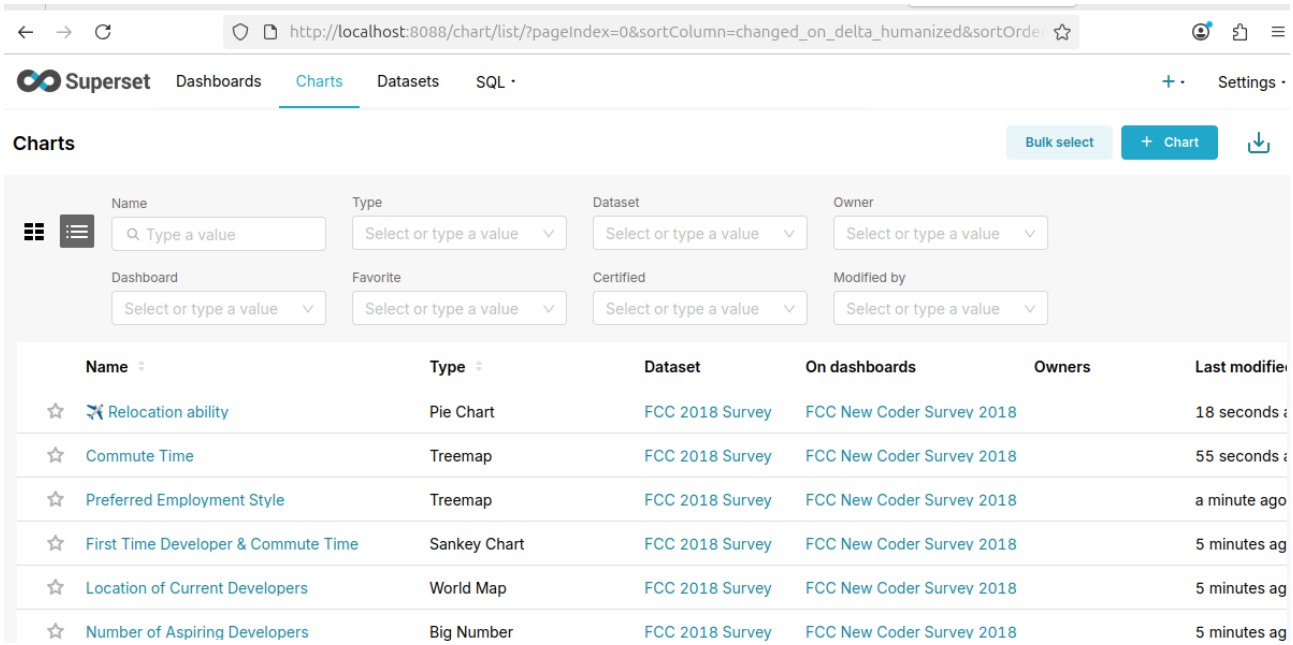
```
 permissions.
2025-06-30 11:33:06,425:INFO:superset.security.manager:Cleaning faulty perms
(venv) ardent@ardent:~/Workspace/superset$ superset run -p 8088 --with-threads -
-reload --debugger
```

10. Go to webbrowser and type localhost:8088. Enter the username and password you set earlier



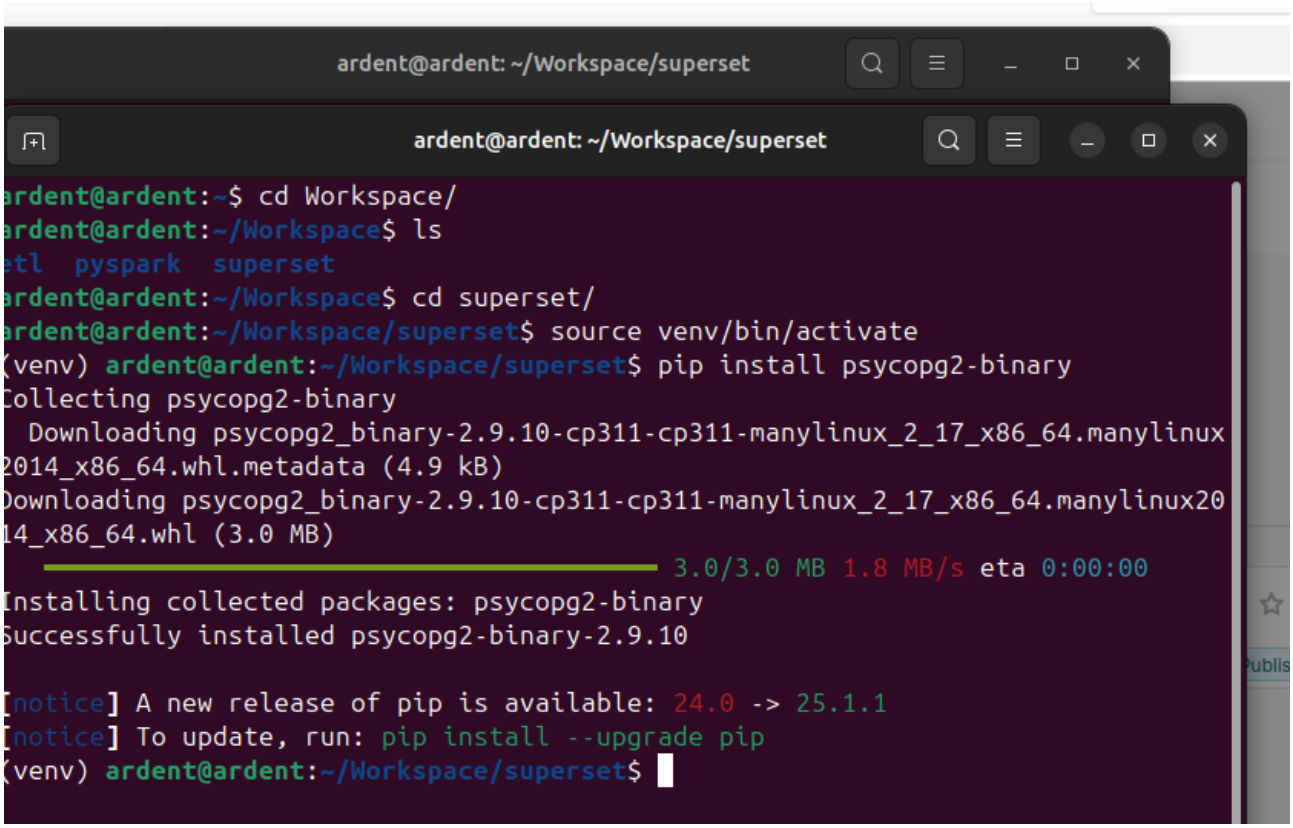
11. Go to dashboard section and charts section and view default charts. Explore using different options



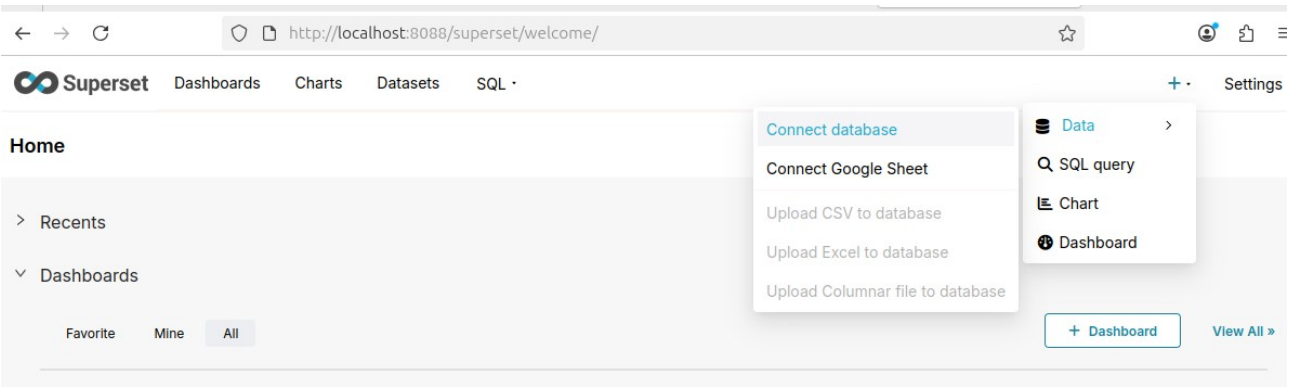


12. Let us connect our local postgres server to superset

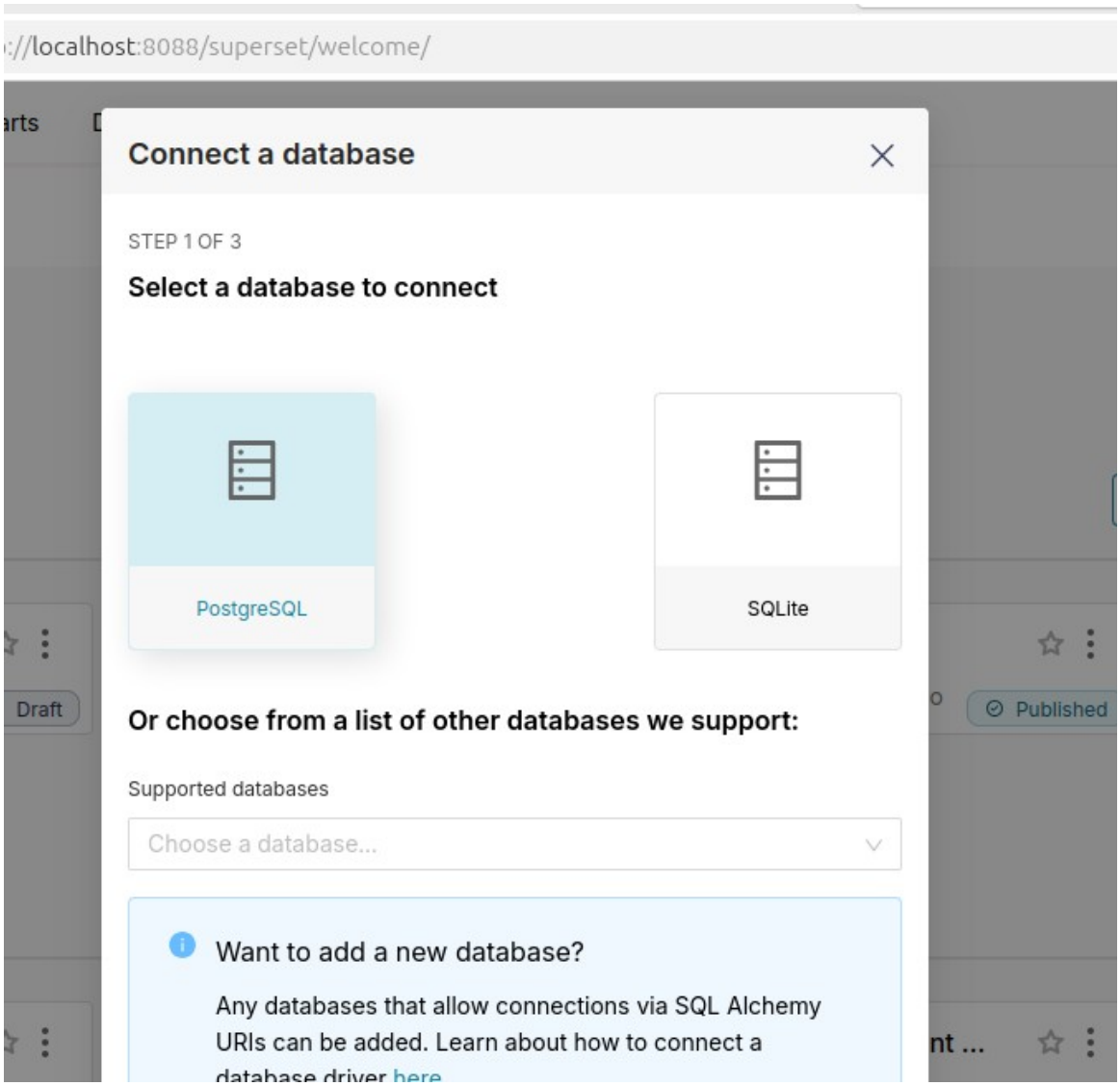
First let us install pyscopg2-binary package in the same virtual environment we installed superset
You might need to restart the superset server if its already running



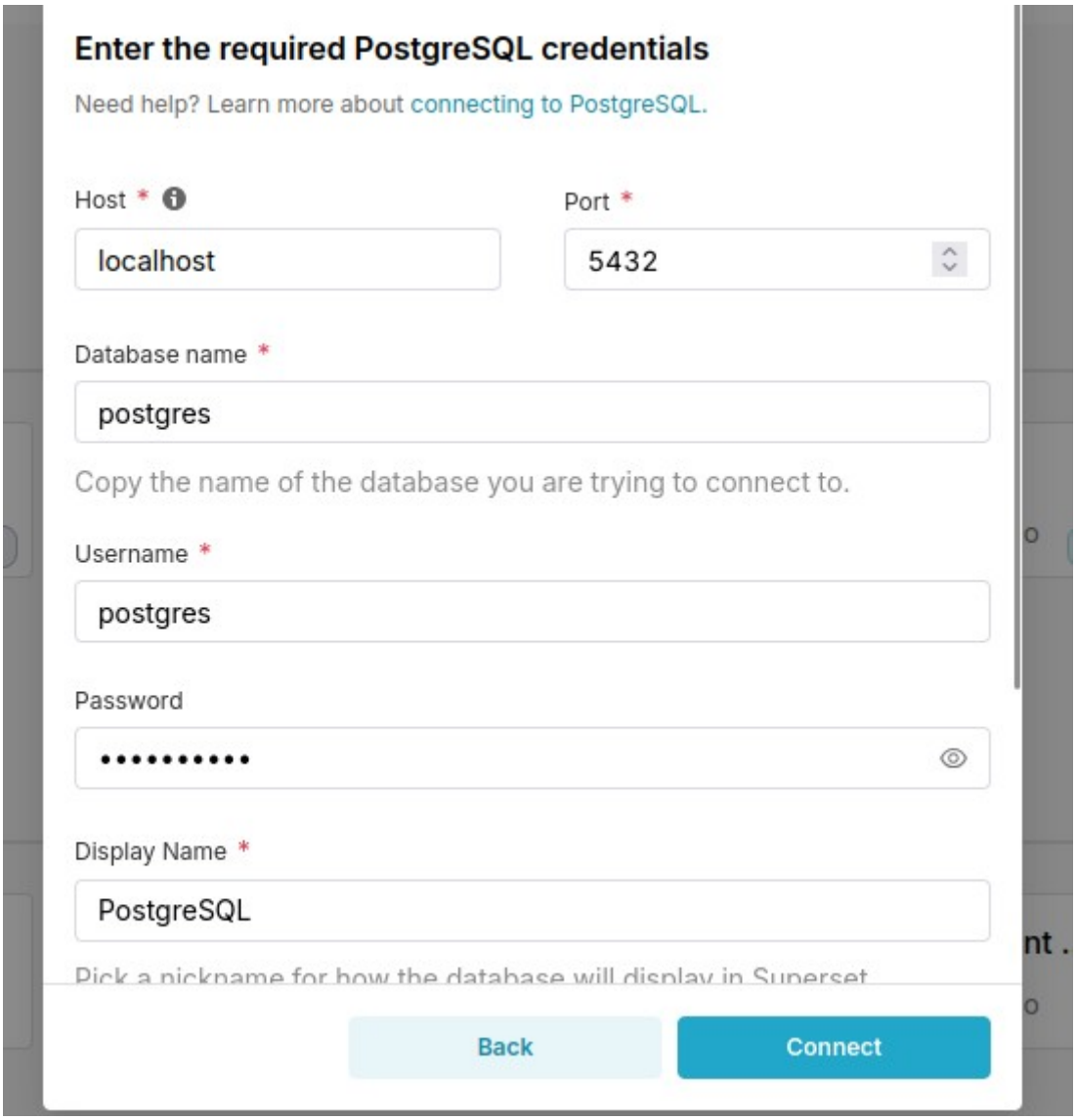
In the top right corner click the + icon then Data > Connect database



Select postgres from the option

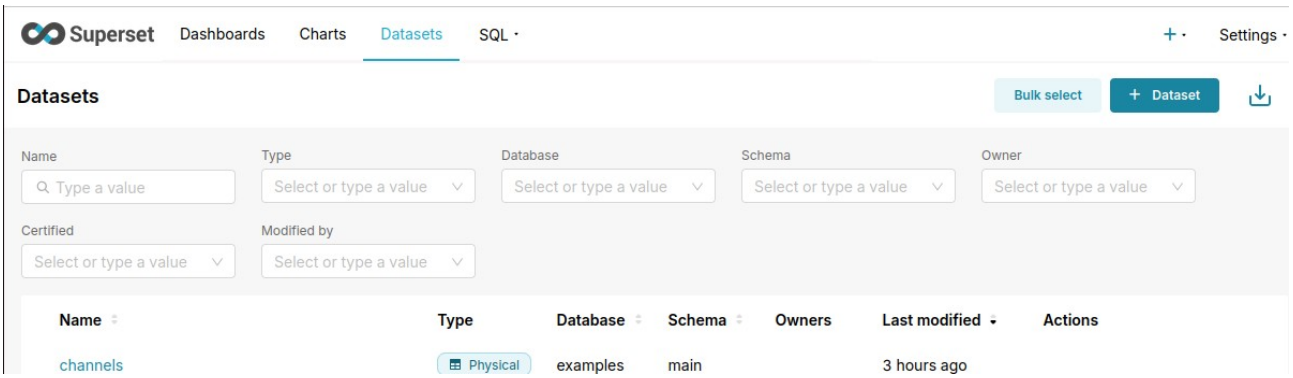


Enter your credentials and press connect

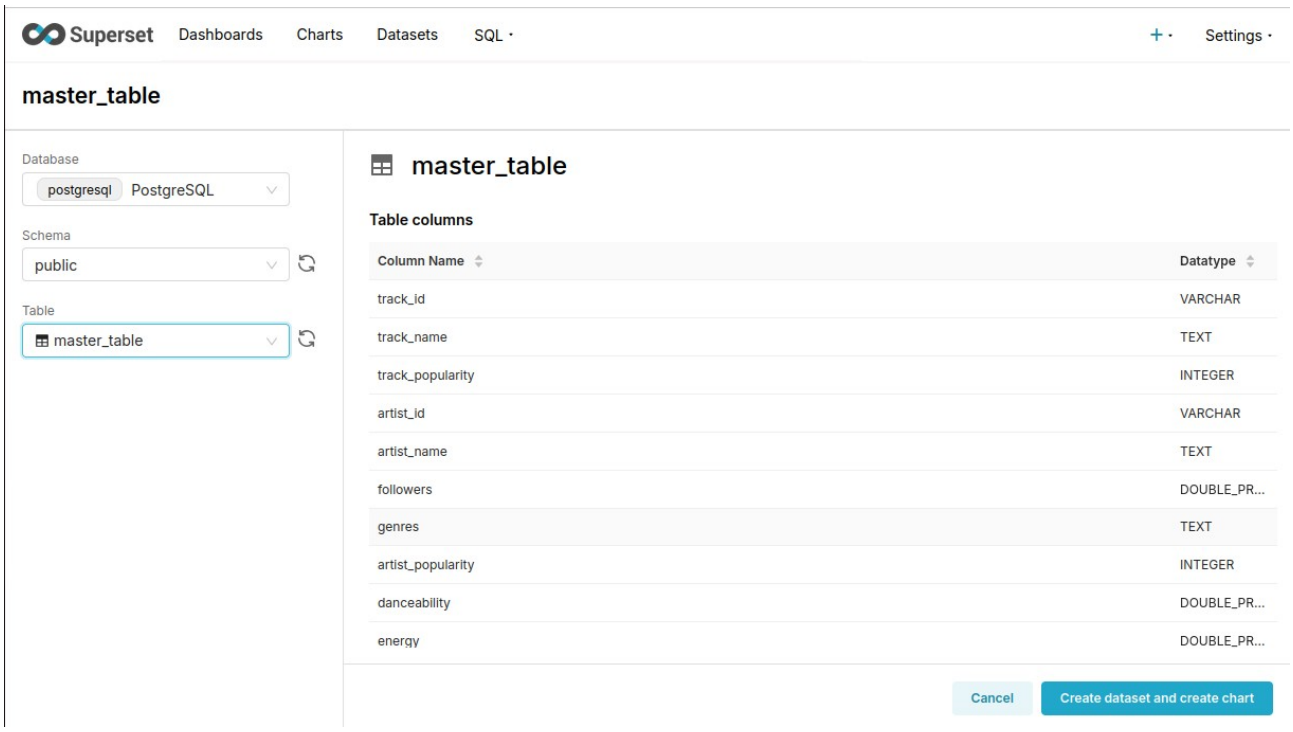


13. Create a dataset out of master table

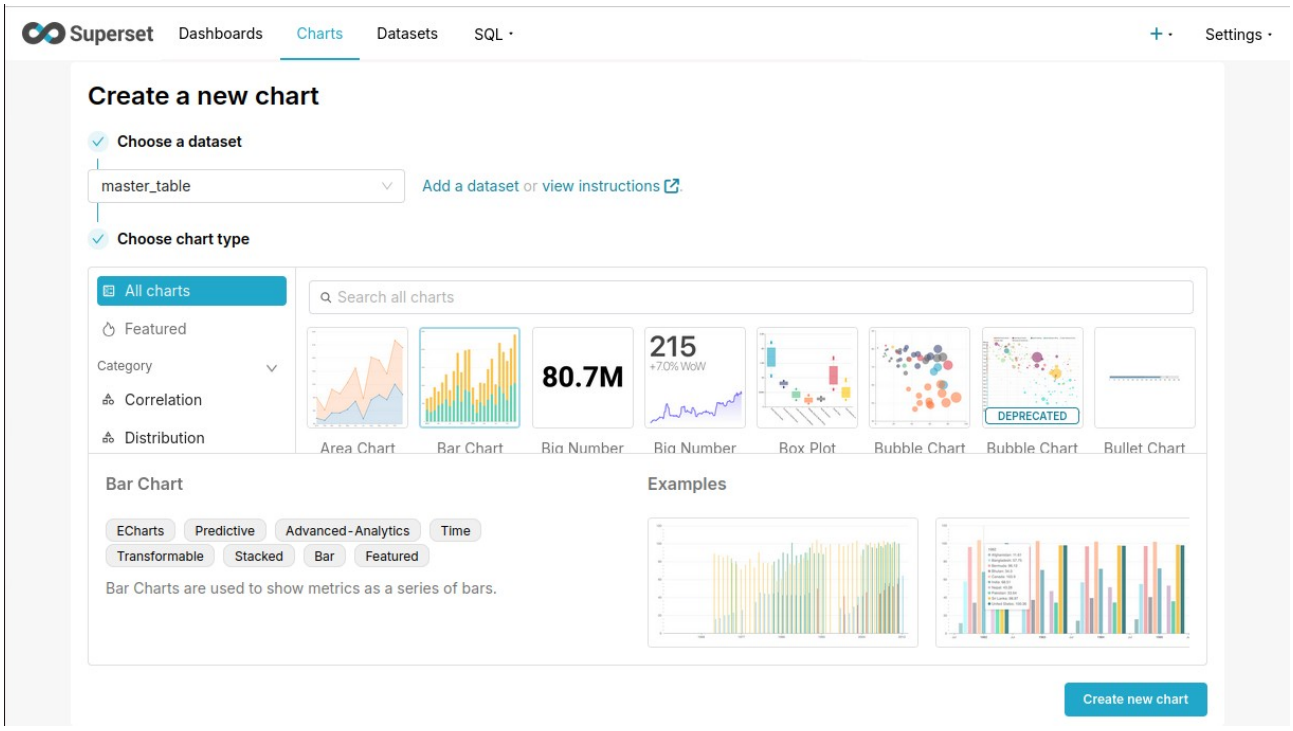
After database connection has been created go to dataset tab and click on +Dataset button



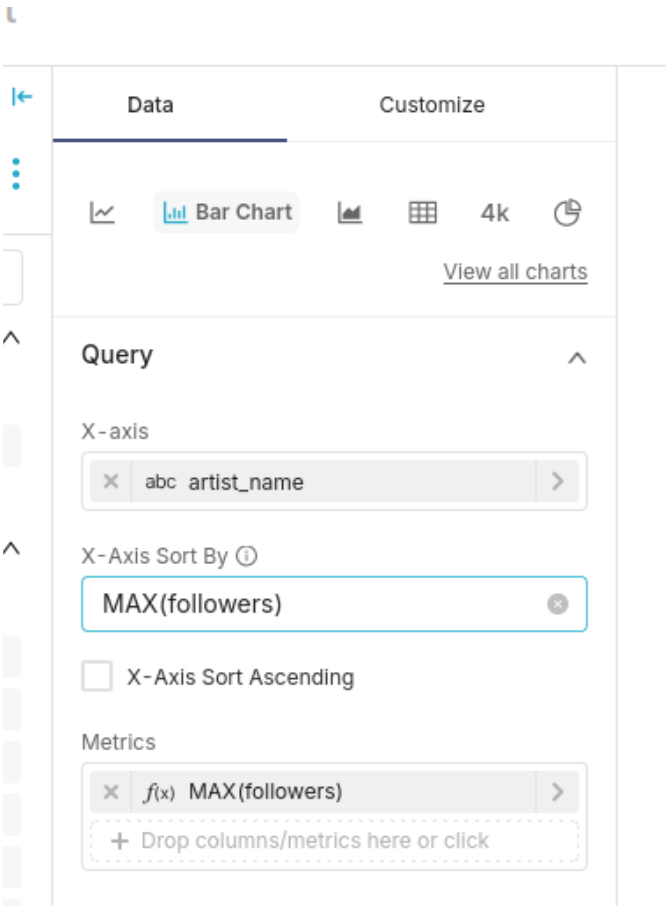
Then select your connection, the public schema, and master table and create dataset and create charts



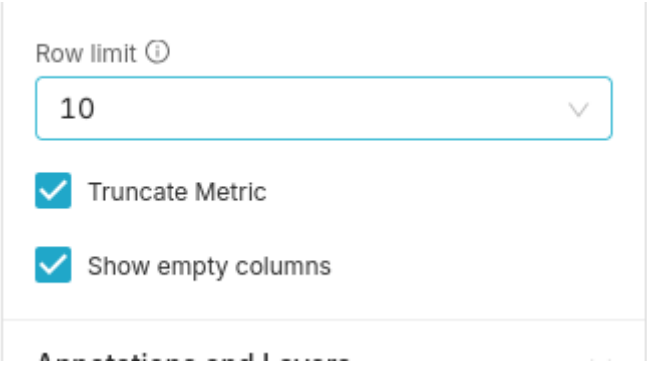
Then you will be redirected to charts section. Let us select bar chart first



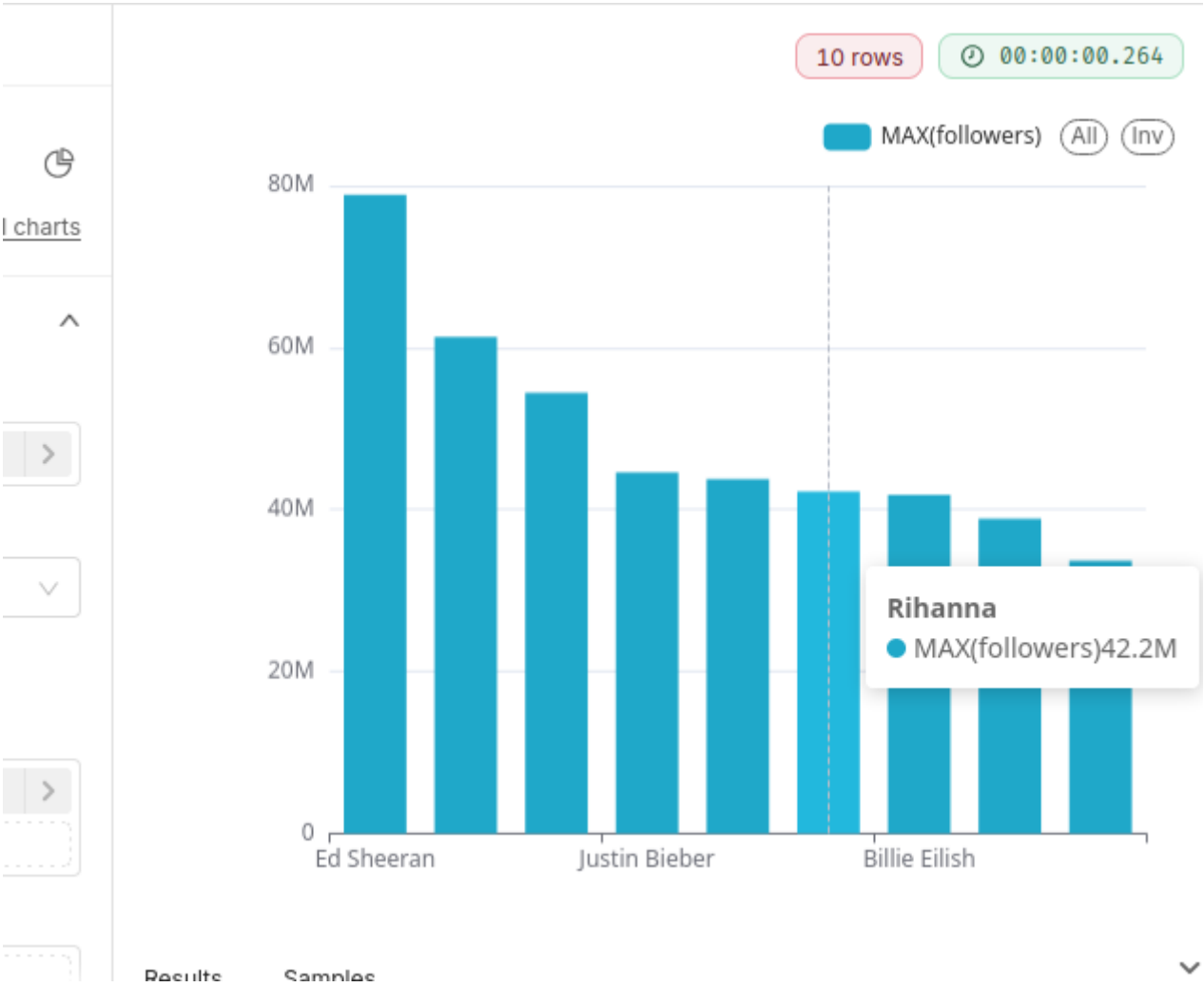
Go to x-axis and select artist_name
Go to metrics and select select followers on the column and max on the aggregate. You might have to go to the simple tab
In X-axis sort-by select max_followers and disable the ascending checkbox as we want ordering in the descending order



Scroll down to row limit and select 10

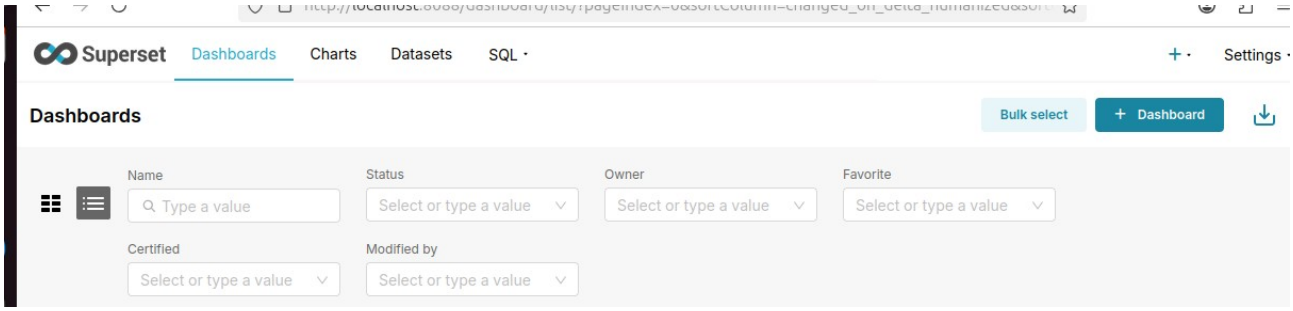


Then create chart you should get something like below:



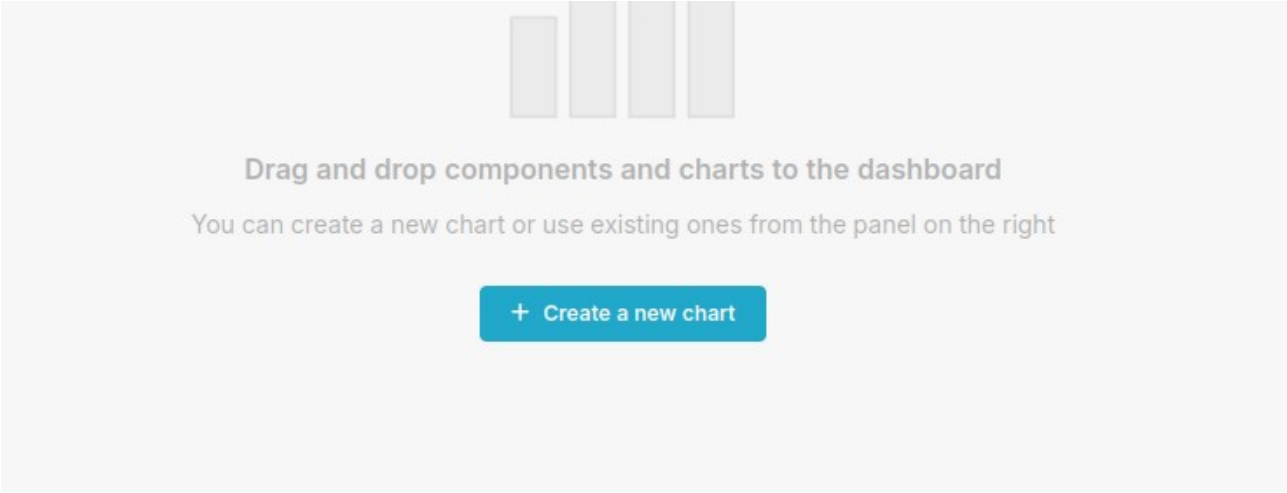
14. Create a dashboard from the master table

Go to dashboard tab and click the + Dashboard button

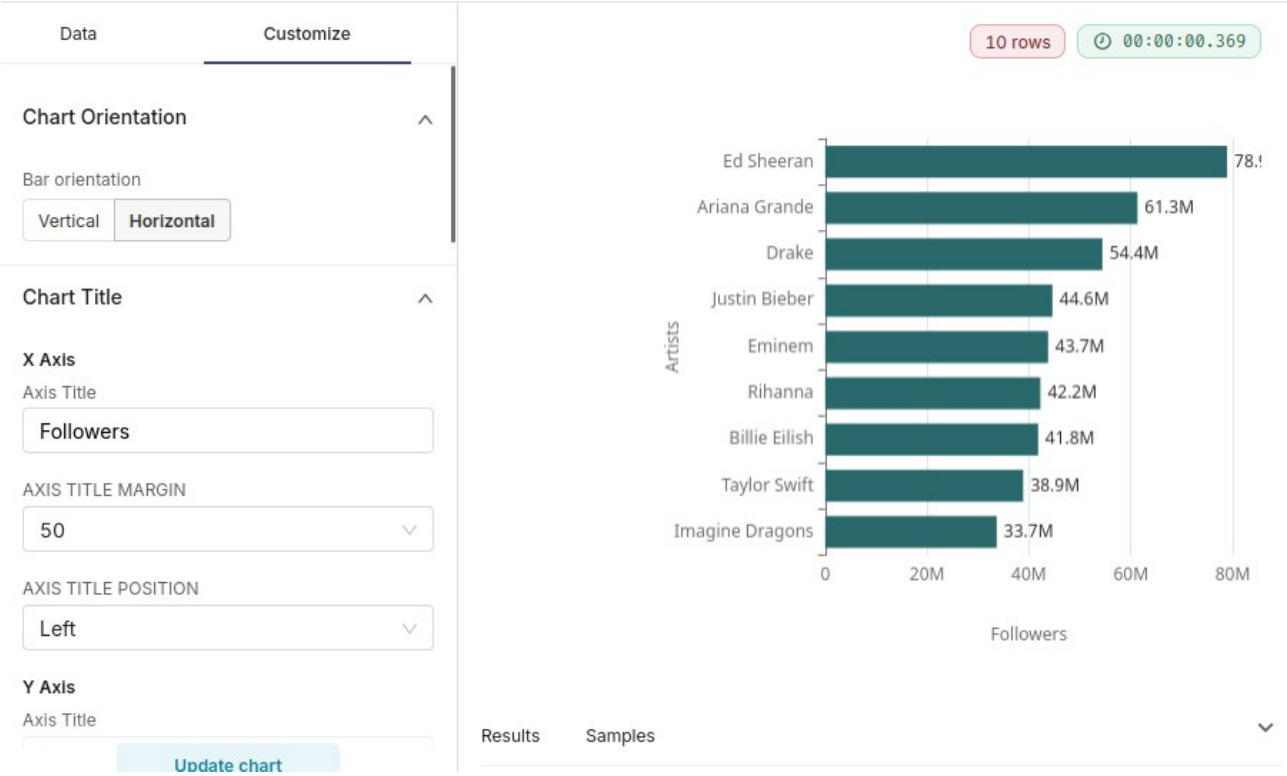


Set the name to Spotify dashboard

Click on Create new Chart

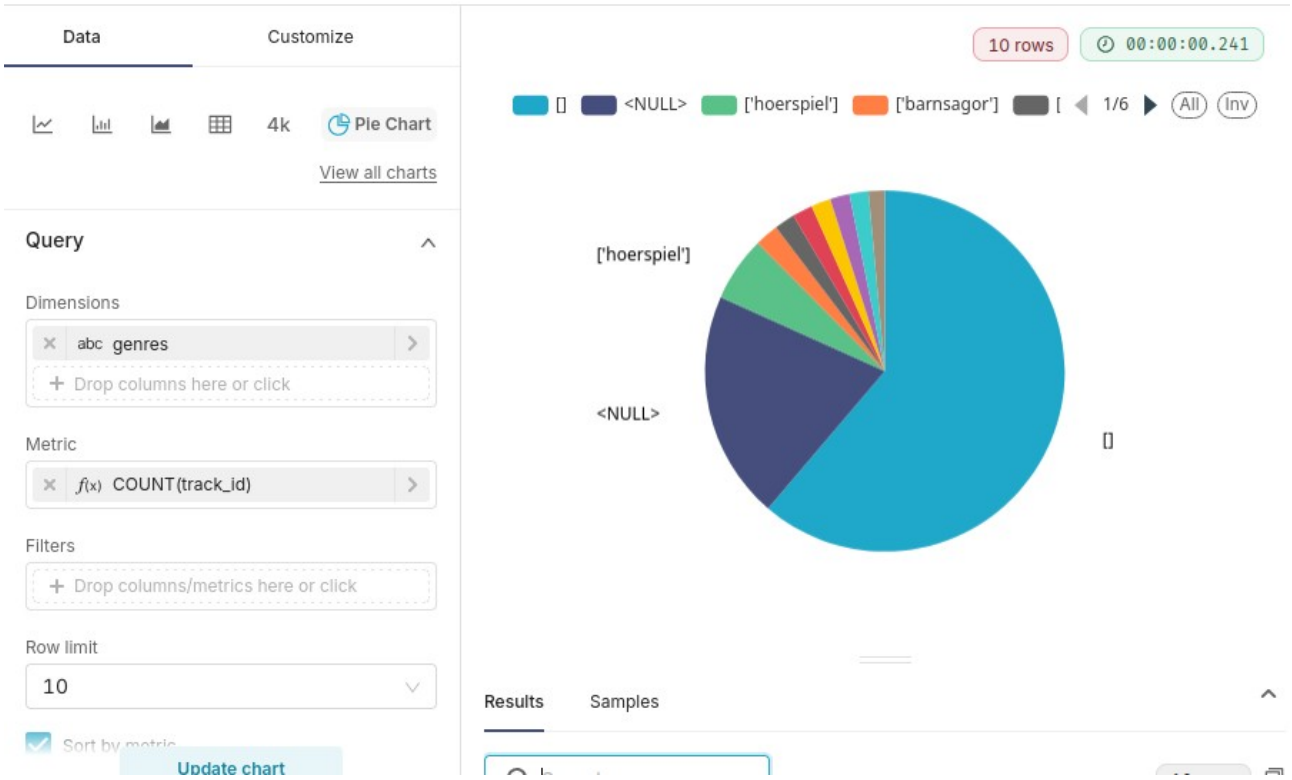


Choose master_table as dataset and Create the previously created bar chart. Go to Customize tab to change colors, give names to axes and so on.

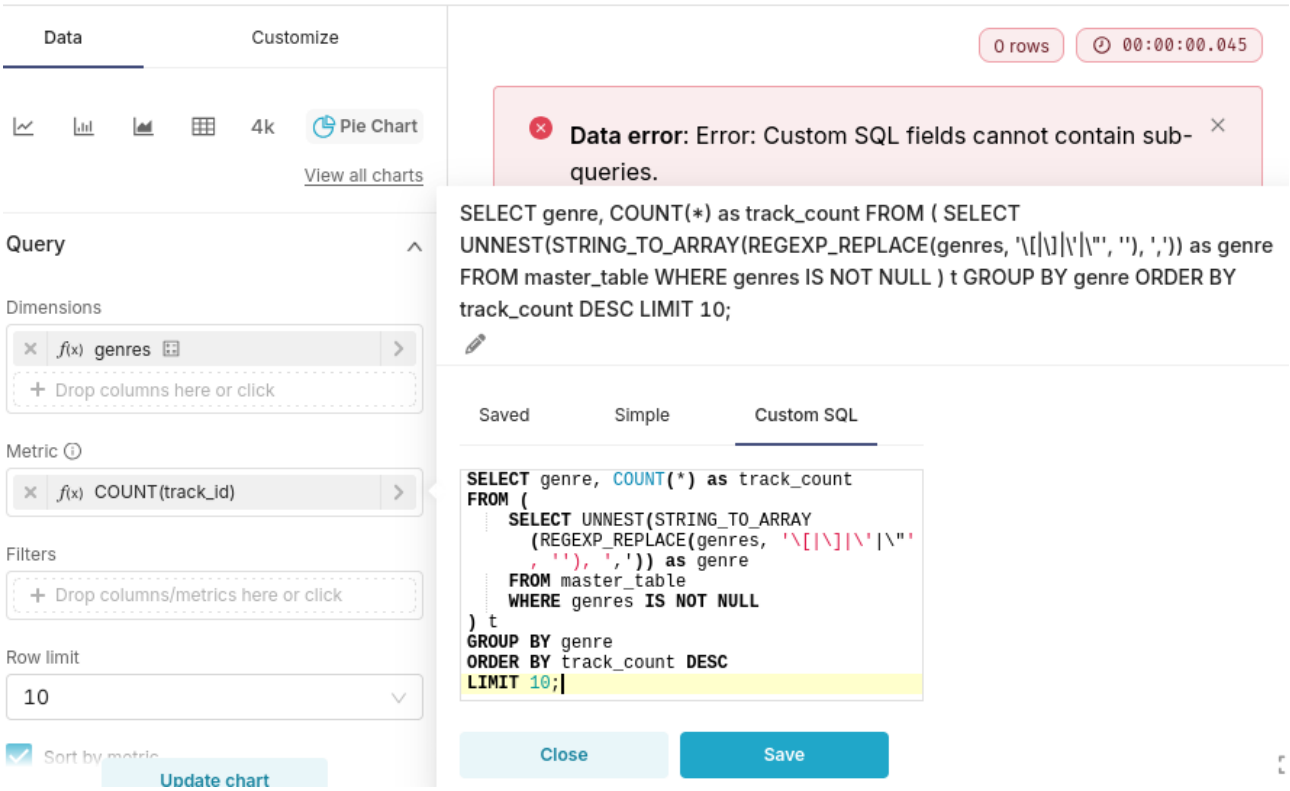


Make sure you save it to the dashboard created earlier. Explore other charts and different ways to querying

In the same dashboard we created last week let us try to add a pie chart. This pie chart will have genre distribution. If you look at below image you should see two main things. First the genre column is array of string, next some of the genres are NULL. This could have been removed from the processing but we do not want to loose track record just because it does not have a genre attached to it. Instead we can use the sql to query only required data and also unnest the array



We can try the following query first but latest version of apache has subquery support disabled by default so we need to modify the config file.



15. Update the config file

Let us navigate the virtual environment folder to find the superset config file:

```
ardent@ardent: ~/Workspace/superset/venv/lib/python3.11/site-package...
ardent@ardent:~$ cd Workspace/superset/
ardent@ardent:~/Workspace/superset$ ls
venv
ardent@ardent:~/Workspace/superset$ cd venv/lib/python3.11/site-packages/superse
t/
ardent@ardent:~/Workspace/superset/venv/lib/python3.11/site-packages/superset$
```

You can also do this using the GUI file explorer.

Open the config.py file in the text editor and change the ALLOW_ADHOC_SUBQUERY to True from False

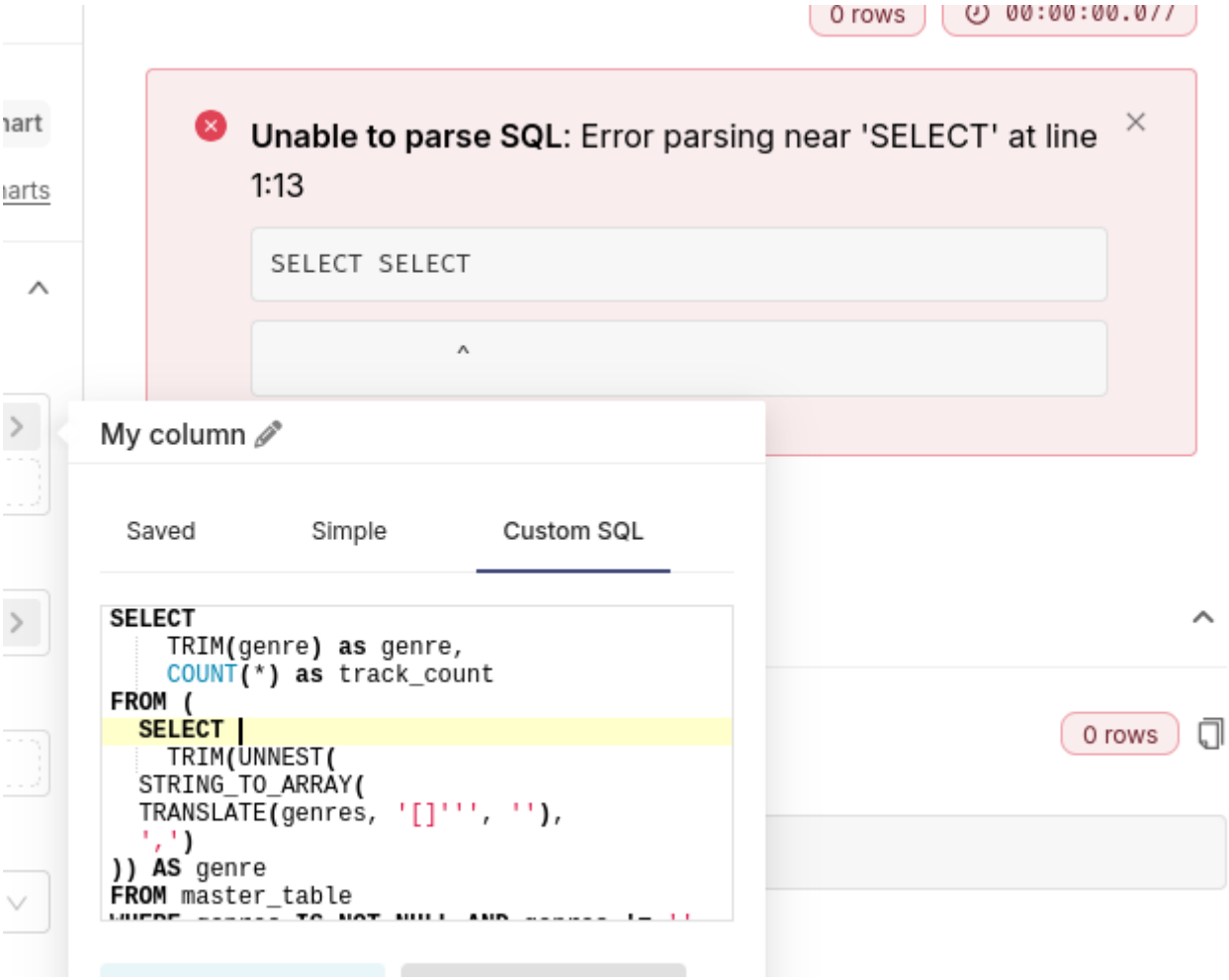
```
venv
ardent@ardent:~/Workspace/superset$ cd venv/lib/python3.11/site-packages/superset/
ardent@ardent:~/Workspace/superset/venv/lib/python3.11/site-packages/superset$ vim config.py
ardent@ardent:~/Workspace/superset/venv/lib/python3.11/site-packages/superset$
```

```
    "ALERTS_ATTACH_REPORTS": True,
    # Allow users to export full CSV of table viz type.
    # This could cause the server to run out of memory or compute.
    "ALLOW_FULL_CSV_EXPORT": False,
    "ALLOW_ADHOC_SUBQUERY": True,
    "USE_ANALOGOUS_COLORS": False,
    # Apply RLS rules to SQL Lab queries. This requires parsing and manipulating the
    # query, and might break queries and/or allow users to bypass RLS. Use with care
    "RLS_IN_SQLLAB": False,
```

Restart the superset server to update the changes. Press CTRL+C to stop the execution then rerun the command to start the server

```
2025-06-30 16:21:28,989:INFO:werkzeug: * Restarting with stat
2025-06-30 16:21:30,619:WARNING:werkzeug: * Debugger is active!
2025-06-30 16:21:30,619:INFO:werkzeug: * Debugger PIN: 449-610-761
^C(venv) ardent@ardent:~/Workspace/superset$ superset run -p 8088 --with-threads
 --reload --debugger
```

Go to dashboard tab > Select the newly created Spotify Dashboard > Click Edit Dashboard button > Create New Chart button



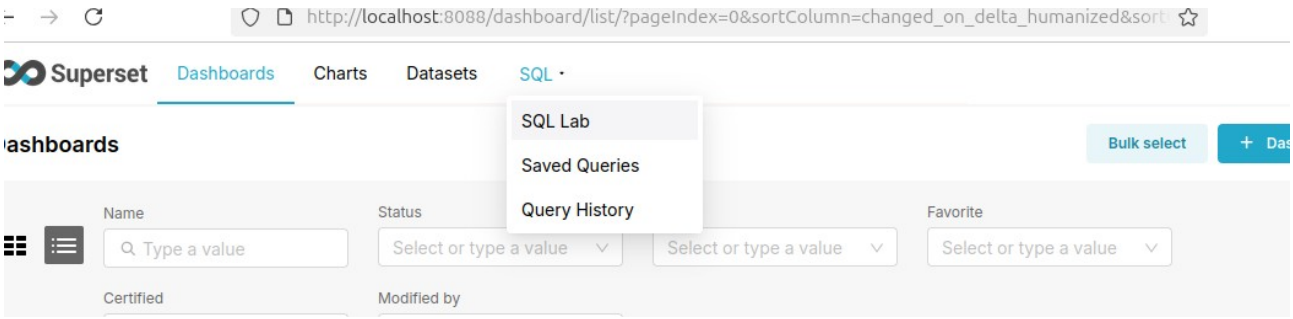
When trying to run the query after allowing subqueries we get a different error. The syntax for Custom Sql is not the same as our default SQL

We need to go to the SQL lab and test our query first to verify that it is correct. Do this every time you need to write a custom query of your own. Use the CUSTOM SQL tab only when you want to work with expressions not complete query

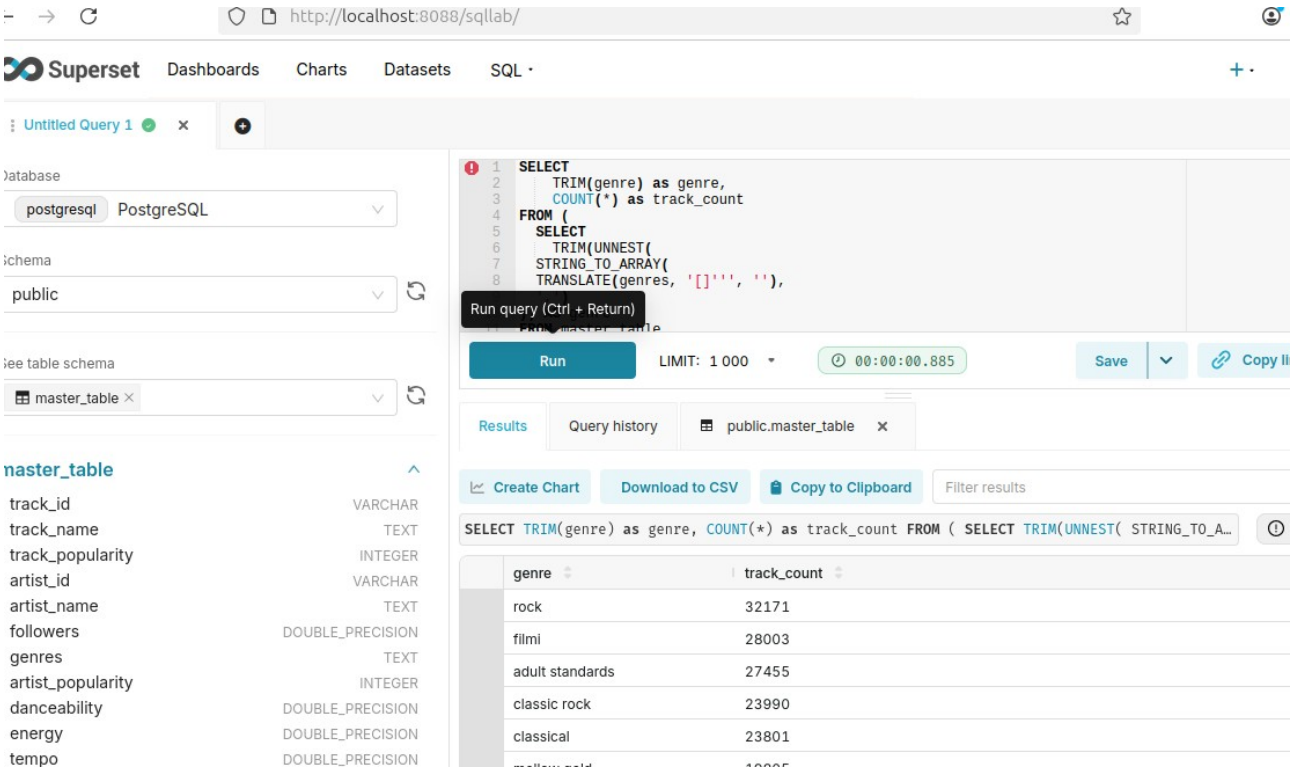
```
SELECT
   TRIM(genre) as genre,
   COUNT(*) as track_count
FROM (
         SELECT
   TRIM(UNNEST(
         STRING_TO_ARRAY(
         TRANSLATE(genres, '[]''', ''),
         ',')
)) AS genre
FROM master_table
WHERE genres IS NOT NULL AND genres != '' AND genres != '[]'
```

```
) AS sub
GROUP BY TRIM(genre)
ORDER BY track_count DESC
LIMIT 10;
```

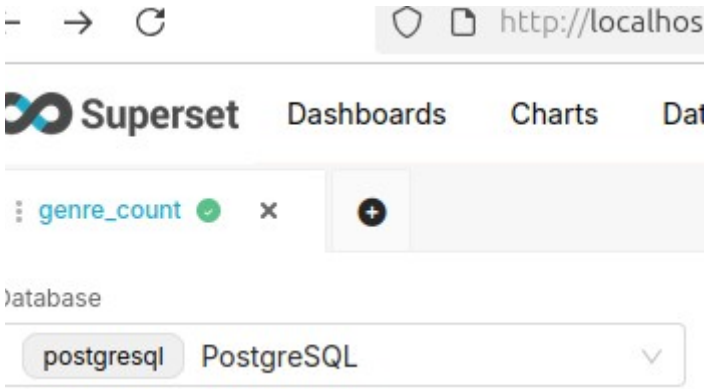
Use the above query but understand what it is doing. Again it is best if you can modify the ETL pipeline to cast the column to array type to solve the issue partially beforehand



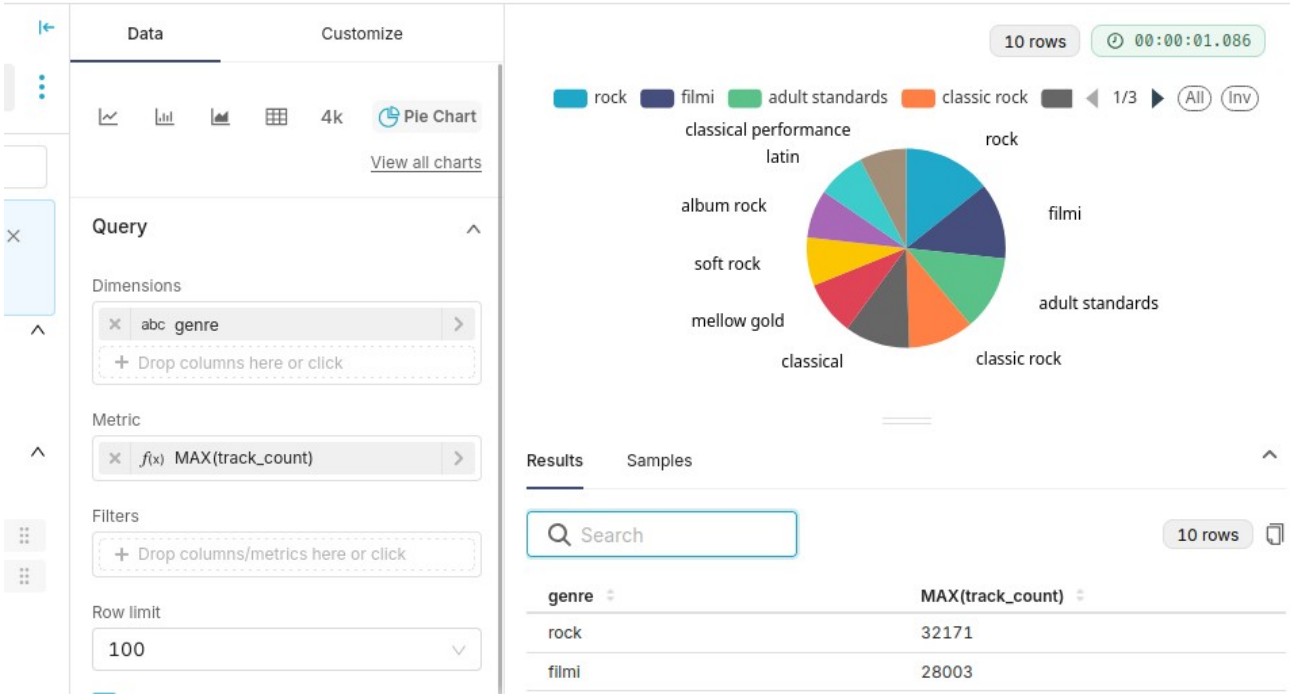
Select your postgresql, schemas as public, table as master_table and paste the query and run. You should get the desired results.



Change the name to something you would recognize from the tab by clicking 3 vertical dots



Then Select the Create Chart button right above the result



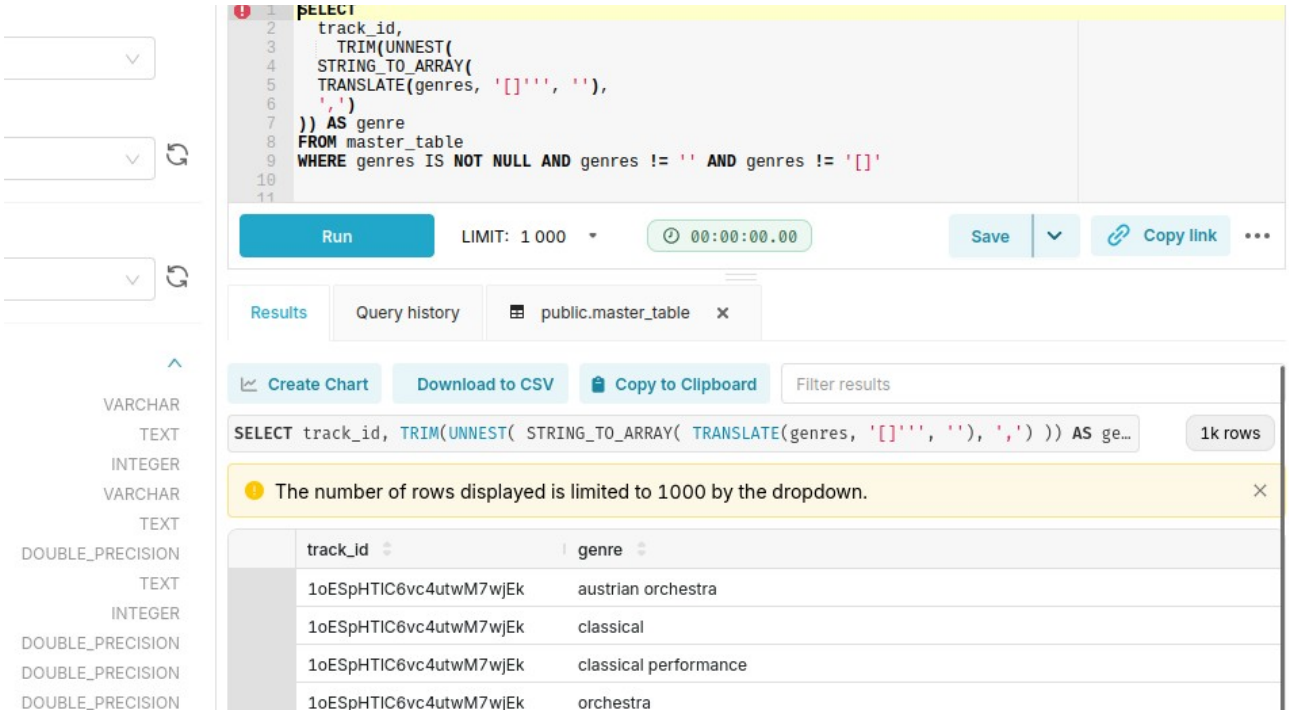
Select Pie Chart and create the chart. If you have noticed the Metric Field compulsorily requires a aggregate function to run. In our query we have already aggregated the data. So running any of the aggregation gives us the required result but this is not optimal. Let us explore this in a different way.

One of the option is to go to the sql lab and update the query to not perform aggregate by default

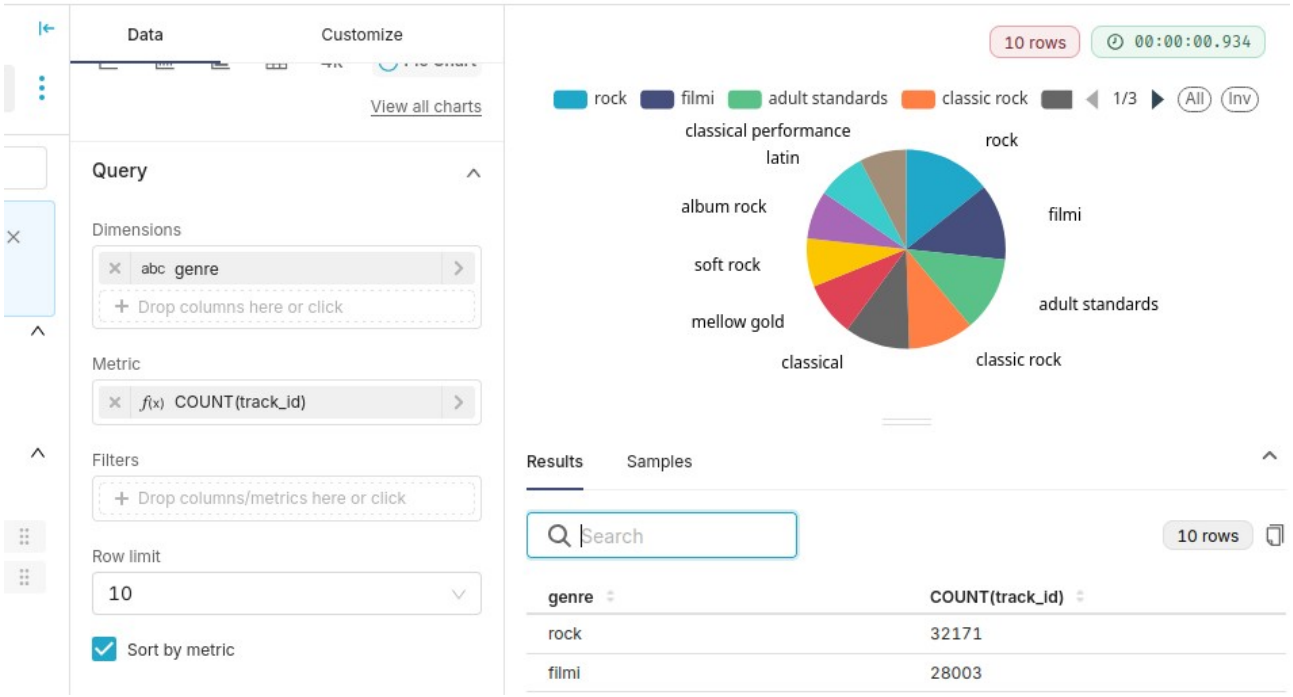
```
SELECT
  track_id,
    TRIM(UNNEST(
          STRING_TO_ARRAY(
```

```
        TRANSLATE(genres, '[]''', ''),
        ',')
)) AS genre
FROM master_table
WHERE genres IS NOT NULL AND genres != '' AND genres != '[]'
```

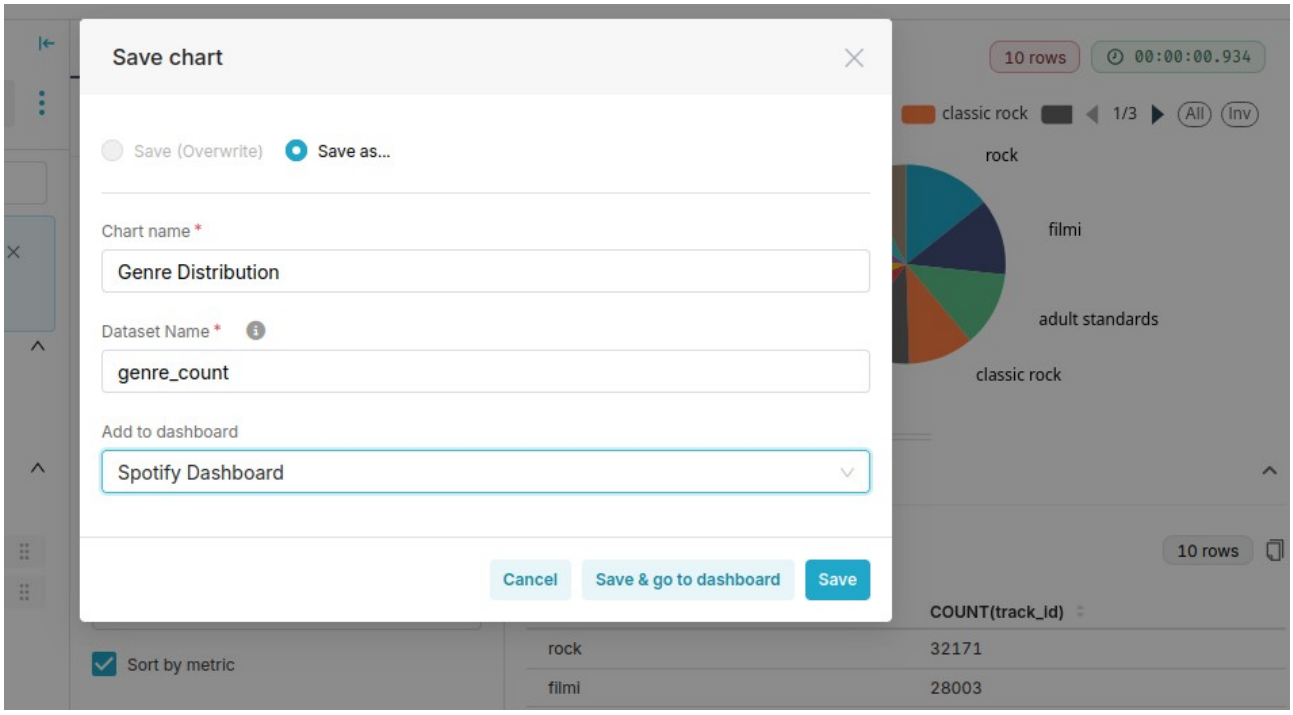
This query will just convert text to array and select track_id along with it as well.



Click Create Chart button.

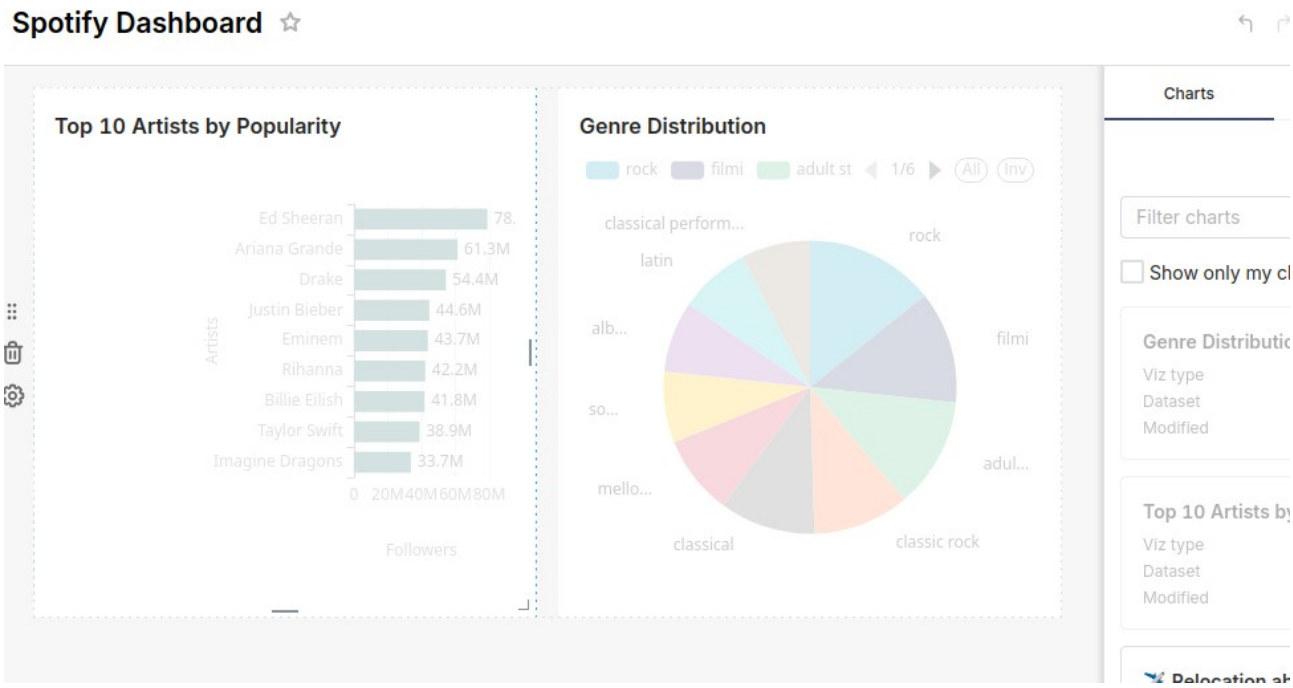


Select Pie Chart. Now we can select the genre tab and perform count on track_id.
There are few other options. Like creating a view or a table in database. Explore them on your own.



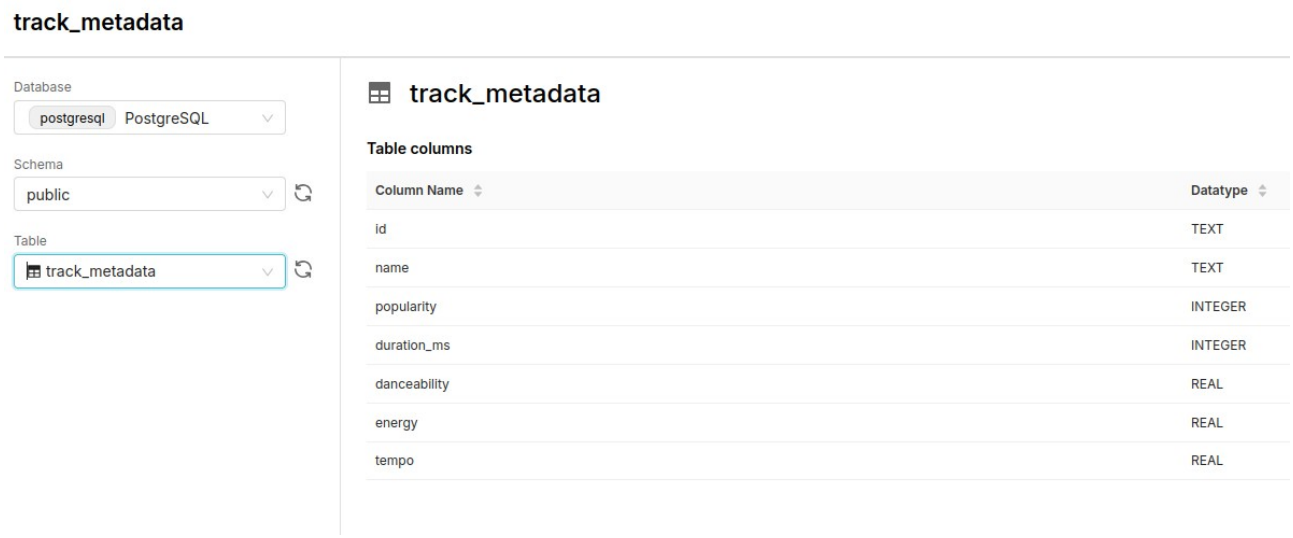
Save the chart. Make sure you choose add to dashboard and select your dashboard

In the dashboard with Edit Dashboard on we can drag and drop charts at different position. We can also resize them.

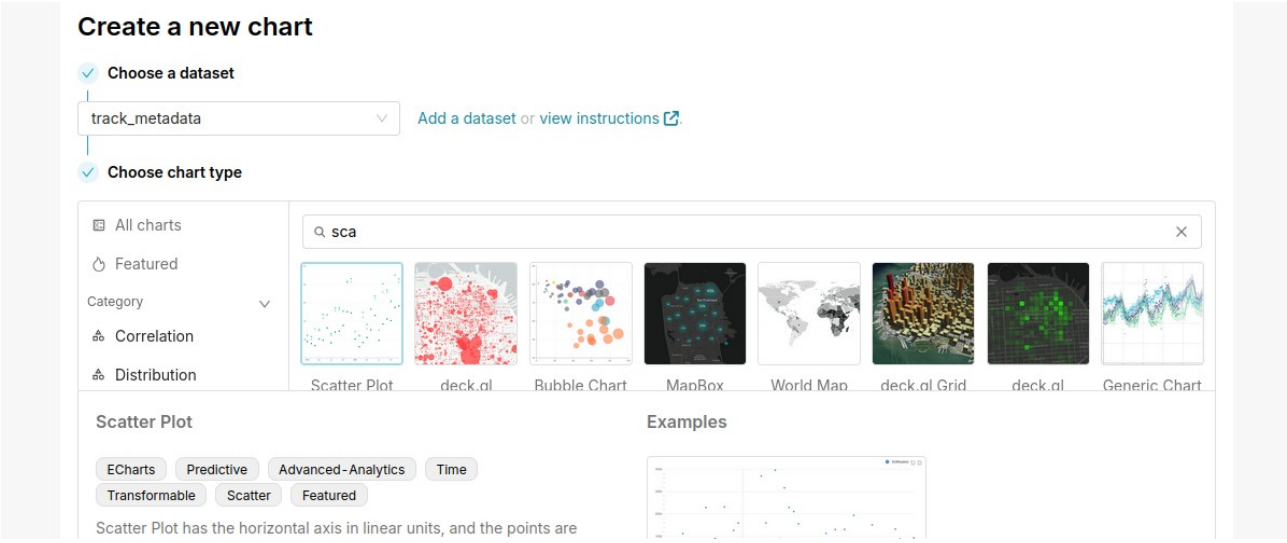


16. Let us create a scatter plot show if danceability influences popularity i.e., are popular tracks more danceable

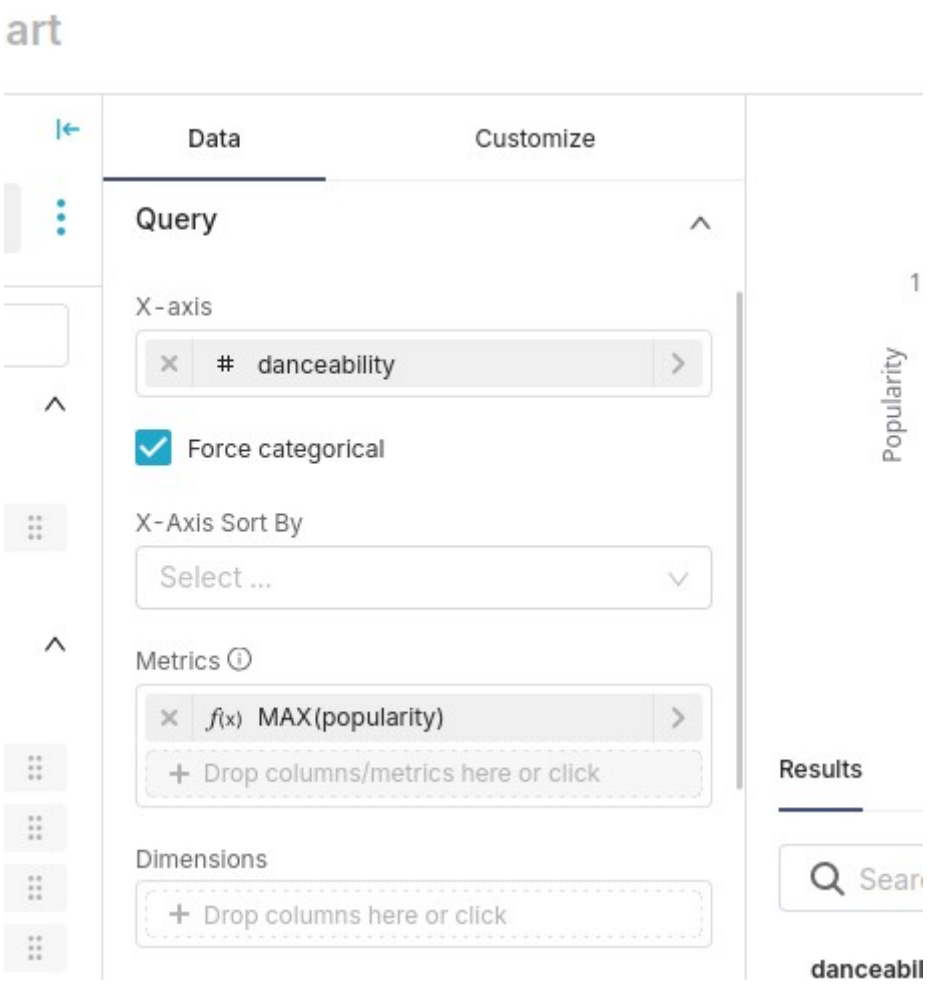
Create a new dataset with track_metadata as the table



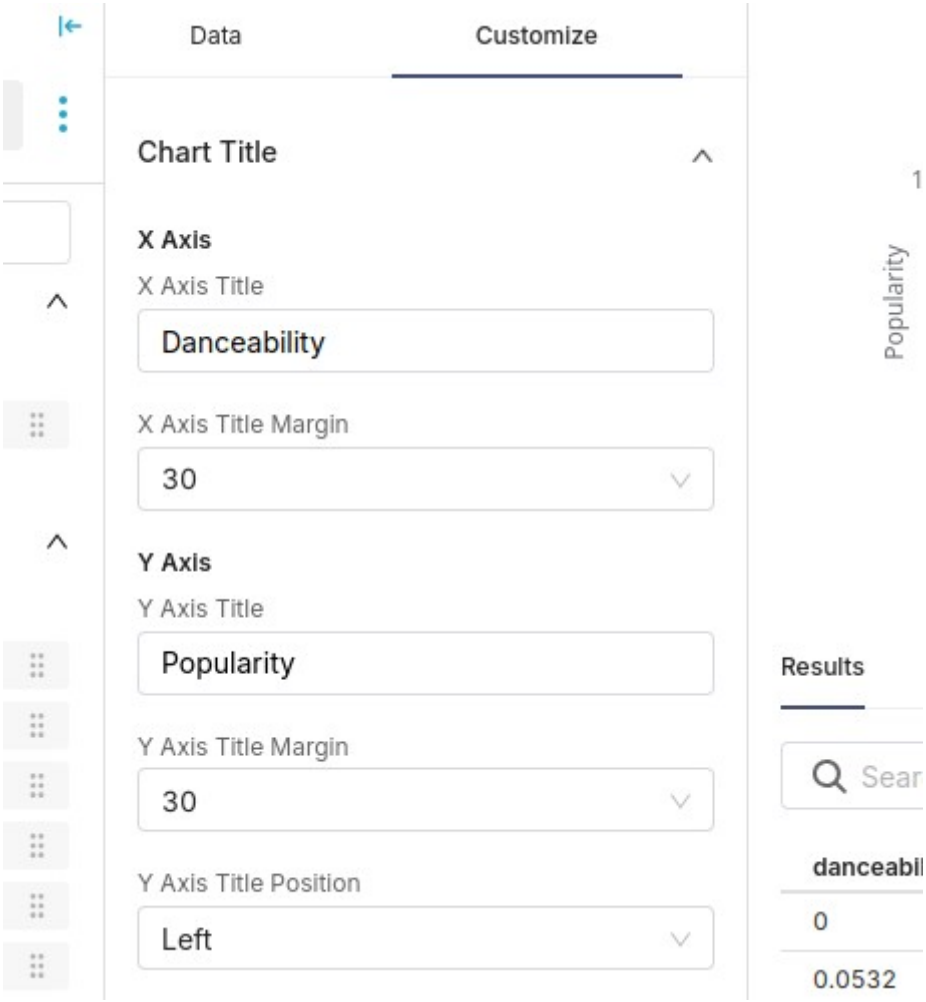
In the create chart section. Make sure track_metadata is selected as dataset and scatter plot as chart type and click create new chart



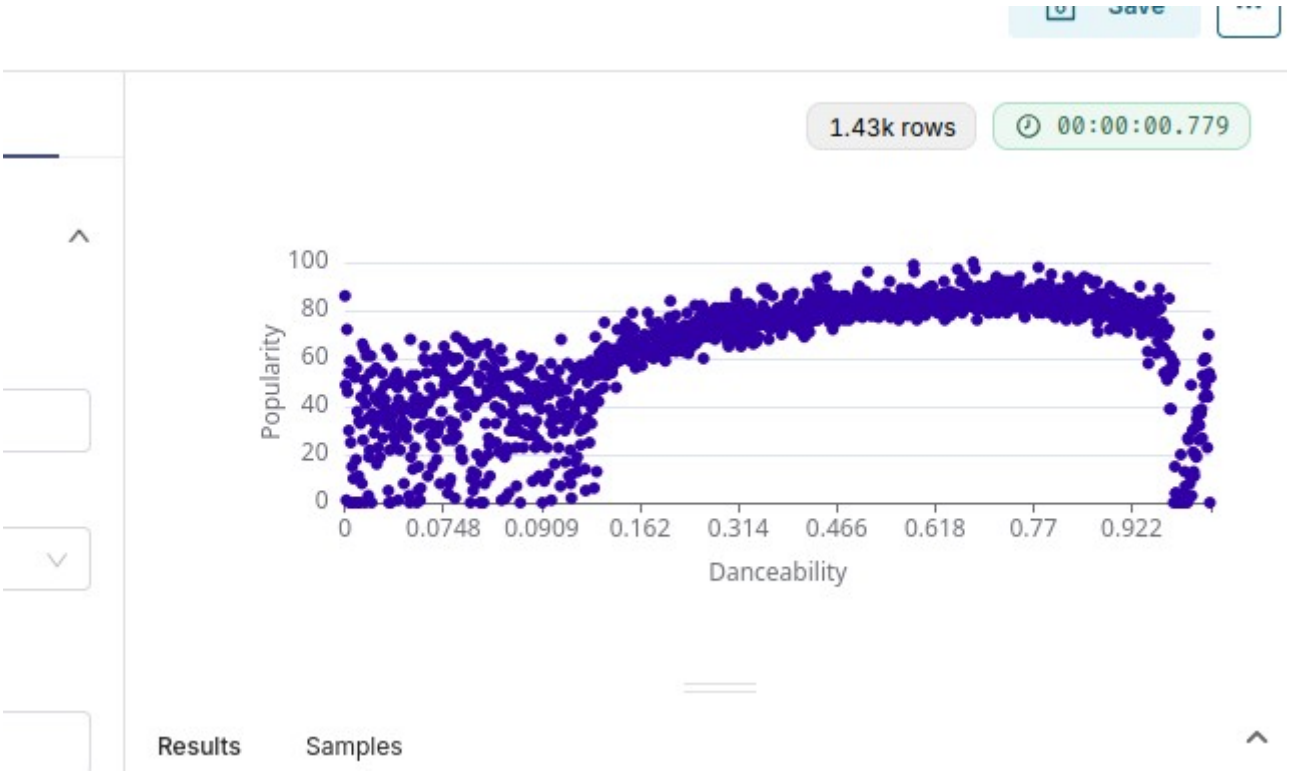
Select danceability as x-axis and max(popularity) as metrics (y-axis). Make sure Force categorical is selected



Make sure you select 50,000 rows and untick the show empty column checkbox so that all proper data is displayed .Also in the customize section give proper naming to x and y axis and set the margins as per your requirement.

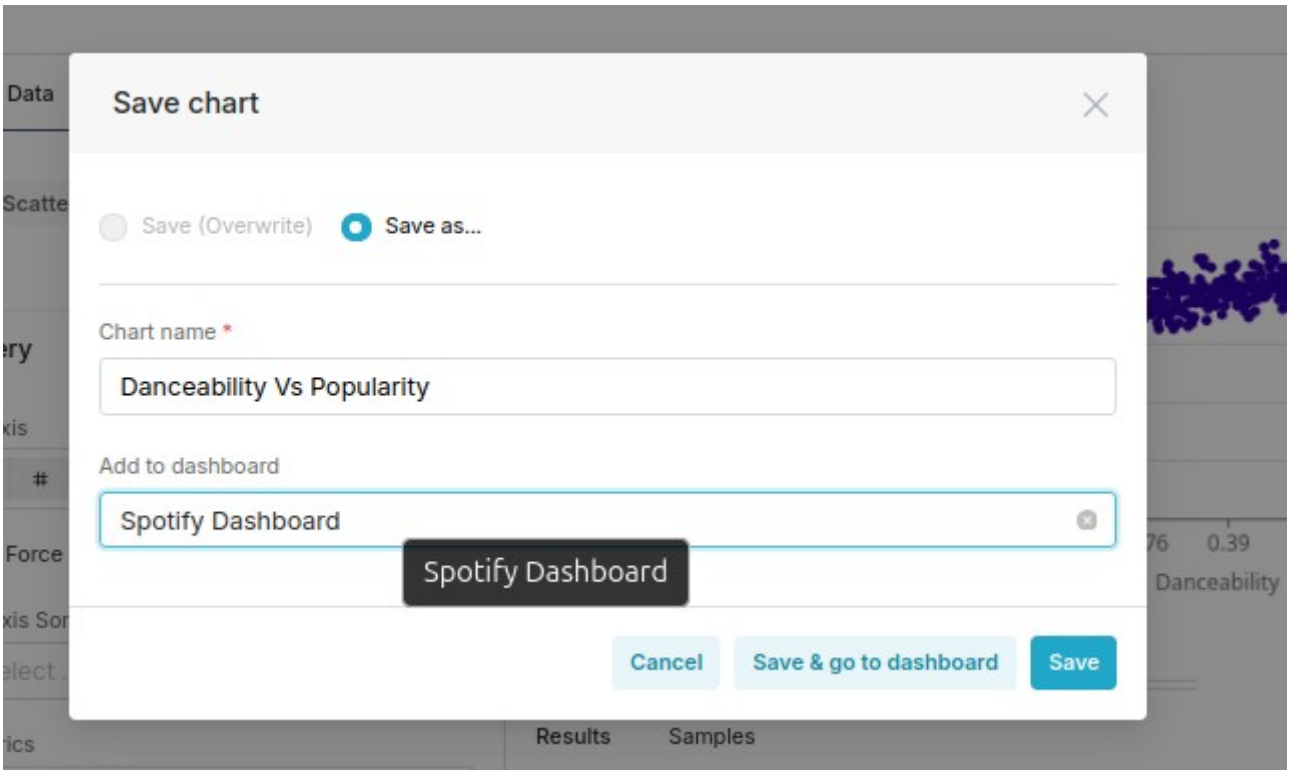


The chart should be somewhat similar to below:



The above graph tells us that. Popularity does increase with danceability but only upto a threshold value of around 0.9 and after that it falls off. Maybe because the music is too loud at that point. This could also be noise in the data. Moreover, bubble chart with energy as size parameter might provide insights into that. Explore this on your own.

17. Save the chart and make sure you add it to the dashboard while saving. Then Click Save and go to Dashboard



Go to Edit Dashboard and rearrange the charts as you like using drag and drop and resize: