CREATE DATABASE

```
-- DROP DATABASE IF EXISTS spotify_project;

CREATE DATABASE spotify_project
    WITH
    OWNER = postgres
    ENCODING = 'UTF8'
    LC_COLLATE = 'C'
    LC_CTYPE = 'C'
    TABLESPACE = pg_default
    CONNECTION LIMIT = -1
    IS_TEMPLATE = False;

DROP TABLE Master;

CREATE TABLE Master (
  album_id VARCHAR(50),
  album_name TEXT NOT NULL,
  album_release_date DATE,
  album_album_type TEXT,
  album_available_markets TEXT,
  track_duration_ms INTEGER,
  track_id VARCHAR(50),
  track_name TEXT NOT NULL,
  track_popularity INTEGER,
  track_track_number INTEGER,
  track_available_markets TEXT,
  artists_id VARCHAR(50),
  artists_name TEXT NOT NULL,
  artists_spotify_url TEXT,
  available_markets TEXT
);
```

IMPORT DATA

Manually imported Data by right clicking on ‘master’ table and selecting the Import/Export feature.

CREATE TABLES

```
drop table Album;
create table Album as SELECT DISTINCT album_id, album_name, album_release_date,
album_album_type, album_available_markets
```

```
FROM Master;

drop table Track;
create table Track as SELECT DISTINCT track_id, track_name, track_popularity,
track_duration_ms, track_available_markets
FROM Master;

drop table artists;
create table artists as SELECT DISTINCT artists_id, artists_name, artists_spotify_url
FROM Master;

drop album_artists_track;
create table album_artists_track as SELECT DISTINCT album_id, artists_id, track_id,
track_track_number,available_markets
FROM Master;

ALTER TABLE album ADD PRIMARY KEY (album_id);
ALTER TABLE artists ADD PRIMARY KEY (artists_id);
ALTER TABLE track ADD PRIMARY KEY (track_id);
ALTER TABLE album_artists_track ADD PRIMARY KEY (album_id, artists_id, track_id);

ALTER TABLE album_artists_track
   ADD CONSTRAINT fk_aat_album FOREIGN KEY (album_id) REFERENCES album
(album_id);
         ADD CONSTRAINT fk_aat_artists FOREIGN KEY (artists_id) REFERENCES artists
(artists_id),
   ADD CONSTRAINT fk_aat_track FOREIGN KEY (track_id) REFERENCES track (track_id);


DROP TABLE Master;
```

PROJECT AND INSIGHTS

According to API guidelines, popularity is calculated by algorithm and is based, in the most part, on the total number of plays the track has had and how recent those plays are.
Let’s look at the songs with high popularity.
How many are there ? Who are the artists that made those songs ?

```
select track_name
from track
where track_popularity > 80;

select count(track_name)
from track
```

```
where track_popularity > 80;

select track_name, artists_name
from track join album_artists_track using (track_id) join artists using(artists_id)
where track_popularity > 80;

select distinct artists_name
from track join album_artists_track using (track_id) join artists using(artists_id)
where track_popularity > 80;
```

–How many artists worked on those 13 songs?

```
select count(distinct artists_name)
from track join album_artists_track using (track_id) join artists using(artists_id)
where track_popularity > 80;
```

–Which song is the most popular? Who made it ? How many songs does this artist have on the playlist ?

```
select track_name
from track join album_artists_track using (track_id) join artists using(artists_id)
where track_popularity > 80
order by track_popularity desc
limit 1;

select artists_name
from track join album_artists_track using (track_id) join artists using(artists_id)
where track_popularity > 80
order by track_popularity desc
limit 1;


select count(track_id)
from album_artists_track
where artists_id = (select artists_id
from track join album_artists_track using (track_id)
where track_popularity > 80
order by track_popularity desc
limit 1);
```

Which artists have the most tracks on that playlist I extracted? Let’s look at the 3 first ones

```
select artists_name, count(track_id) as number_of_songs
from album_artists_track join artists using (artists_id)
group by artists_name
order by number_of_songs desc
limit 3;
```

Featurings

```
select array_agg(artists_name) as featurings
from album_artists_track join artists using (artists_id) join track using (track_id)
where track_id in
(select track_id
from album_artists_track
group by track_id
having count(artists_id) > 1)
group by track_name;
```

DROPPING ALL THE TABLES

```
DROP TABLE album_artists_track;
DROP TABLE album;
DROP TABLE track;
DROP TABLE artists;

DROP DATABASE spotify_project;
```