```
------ BUILDING THE COMBINED DATA SET ------

-- Creating tables in SQL to match data provided in the csv. files,
noting that "region" in song_data is a foreign key from "code" in
country_mapping
DROP TABLE IF EXISTS song_data;
DROP TABLE IF EXISTS country_mapping;

 CREATE TABLE country_mapping (
        name VARCHAR,
        code VARCHAR,
        PRIMARY KEY (code)
 );

 CREATE TABLE song_data (
        position INTEGER,
        track_name VARCHAR,
        artist VARCHAR,
        streams BIGINT,
        url VARCHAR,
        date DATE,
        region VARCHAR,
        FOREIGN KEY (region) REFERENCES country_mapping (code)
 );

-- After importing country_mapping.csv update the country_mapping
table by:
    -- Updating the "code" column to be in lowercase to allow the
"song_data" column "region" foreign key to reference the "code"
column
    -- Inserting a new row/country/code "global" which is in
data.csv but is missing from country_mapping.csv
update country_mapping set code = lower(code);

insert into country_mapping (name, code)
    values ('global', 'global');

-- After importing both csv files, combine the tables based on the
country/region/code.
create table combined_data as
        select * from song_data
        inner join country_mapping on song_data.region =
country_mapping.code;

alter table combined_data
    drop column code;

-- Manually note the time length of this data: 2017 - 2018
select min(date) from combined_data
select max(date) from combined_data



------ TOP 3 RANKING ARTISTS AND SONGS FOR EACH REGION ------
```

```
-- Creating new views that sum total number of streams for artists
and tracks for each region
create view artist_streams as
select
        combined_data.name,
        combined_data.artist,
        sum(combined_data.streams) as "artist_stream_sum" from
combined_data
        group by combined_data.name, combined_data.artist;

create view track_streams as
select
        combined_data.name,
        combined_data.track_name,
        sum(combined_data.streams) as "track_stream_sum" from
combined_data
        group by combined_data.name, combined_data.track_name;

-- Creating tables to list top 3 artists and songs for each region
create table top3_artist as
select * from(
        select
                name,
                artist,
                artist_stream_sum,
                row_number() over (partition by name order by
artist_stream_sum desc) as artist_rank
        from artist_streams) rank
where artist_rank <= 3;

create table top3_track as
select * from(
        select
                name,
                track_name,
                track_stream_sum,
                row_number() over (partition by name order by
track_stream_sum desc) as track_rank
        from track_streams) rank
where track_rank <= 3;



------ SHARED TOP-RANKING ARTISTS OR SONGS ------

-- Creating views to list top 5 artists and songs for each region
create view top5_artist as
select * from(
        select
                name,
                artist,
                artist_stream_sum,
                row_number() over (partition by name order by
```

```
artist_stream_sum desc) as artist_rank
        from artist_streams) rank
where artist_rank <= 5;

create view top5_track as
select * from(
        select
                name,
                track_name,
                track_stream_sum,
                row_number() over (partition by name order by
track_stream_sum desc) as track_rank
        from track_streams) rank
where track_rank <= 5;

-- Creating tables to show no. of countries who share the same top 5
songs or artists
create table shared_track as
        select count(name), track_name
        from top5_track
        group by track_name
        order by count(name) desc;
delete from shared_track where count < 2

create table shared_artist as
    select count(name), artist
    from top5_artist
    group by artist
    Order by count(name) desc;
delete from shared_artist where count < 2
```