|  |  |
| --- | --- |
| select name\_g, count(id\_artist) from genres as g  join genreartist as g2 on g.id = g2.id\_genre  group by name\_g | select genre\_name, count(id\_artist\_ag) from Genres as g  join Artists\_Genres as ag on g.id\_genre = ag.id\_genre\_ag  group by genre\_name |

|  |  |
| --- | --- |
| select count(title\_al) from tracks as t  join albums as a on t.id\_album = a.id  where date\_al >= '20190101' and date\_al <= '20201231' | select count(album\_name) from Tracks as t  join Albums as a on t.id\_album\_t = a.id\_album  where album\_release\_year >= '20190101' and album\_release\_year <= '20201231' |

|  |  |
| --- | --- |
| select title\_al, avg(duration) from tracks as t  join albums as a on t.id\_album = a.id  group by title\_al | select album\_name, avg from Tracks as t |

select name\_ar from artists as a

join artistalbum as a2 on a.id = a2.id\_artist

join albums as a3 on a3.id = a2.id\_album

where date\_al not between '20100101' and '20101231'

group by name\_ar

select title\_col from collections as c

join collectiontracks as c2 on c2.id\_collection = c.id

join tracks as t on c2.id\_track = t.id

join albums as a on t.id\_album = a.id

join artistalbum as a2 on a2.id\_album = a.id

join artists as a3 on a2.id\_artist = a3.id

where name\_ar = 'Jennifer Lopez'

group by title\_col

insert into artistalbum

Values(6, 5)

select title\_al, count(name\_g) from albums as a

join artistalbum as a2 on a.id = a2.id\_album

join artists as a3 on a2.id\_artist = a3.id

join genreartist as g on a3.id = g.id\_artist

join genres as g2 on g2.id = g.id\_genre

group by title\_al

having count(name\_g)>1

select title\_tr from tracks as t

left join collectiontracks as c on t.id = c.id\_track

where id\_collection is null

select name\_ar, duration from tracks as t

join albums a on t.id\_album = a.id

join artistalbum as a2 on a2.id\_album = a.id

join artists as a3 on a2.id\_artist = a3.id

where duration =(select min(duration) from tracks)

select title\_al, count(title\_tr) from tracks as t

join albums as a on t.id\_album = a.id

group by title\_al

select distinct title\_al from albums as a

left join tracks as t on t.id\_album = a.id

where t.id\_album in (

select id\_album from tracks

group by id\_album

having count(id) = (

select count(id)

from tracks

group by id\_album

order by count

limit 1

)

)

CREATE TABLE if not exists genres (

id SERIAL PRIMARY KEY,

name\_g VARCHAR(40) NOT NULL UNIQUE

);

CREATE TABLE if not exists artists (

id SERIAL PRIMARY KEY,

name\_ar VARCHAR(40) NOT NULL

);

CREATE TABLE if not exists GenreArtist (

id\_genre INTEGER references genres(id) NOT NULL,

id\_artist INTEGER references artists(id) NOT NULL,

constraint pk PRIMARY KEY(id\_genre, id\_artist)

);

CREATE TABLE if not exists albums (

id SERIAL PRIMARY KEY,

title\_al VARCHAR(40) NOT NULL,

date\_al DATE not null

);

CREATE TABLE if not exists ArtistAlbum (

id\_artist INTEGER references artists(id) NOT NULL,

id\_album INTEGER references albums(id) NOT NULL,

constraint pk\_ PRIMARY KEY(id\_artist, id\_album)

);

CREATE TABLE if not exists tracks (

id SERIAL PRIMARY KEY,

title\_tr VARCHAR(40) NOT NULL,

duration time NOT NULL,

id\_album INTEGER references albums(id) NOT NULL

);

CREATE TABLE if not exists collections (

id SERIAL PRIMARY KEY,

title\_col VARCHAR(40) NOT NULL,

date\_col DATE NOT NULL

);

CREATE TABLE if not exists CollectionTracks (

id\_collection INTEGER references collections(id) NOT NULL,

id\_track INTEGER references tracks(id) NOT NULL,

constraint pk\_\_ PRIMARY KEY(id\_collection, id\_track)

);

Footer

insert into genres

values(1, 'pop'),

(2, 'rock'),

(3, 'hip-hop'),

(4, 'R&B'),

(5, 'metal');

insert into Artists

values(1, 'Madonna'),

(2, 'Jennifer Lopez'),

(3, 'Maneskin'),

(4, 'Eminem'),

(5, 'Beyonce'),

(6, 'Rihanna'),

(7, 'Metallica'),

(8, 'Kanye West'),

(9, 'Billie Eilish');

insert into genreartist

values(1, 1),

(1, 2),

(1, 9),

(2, 3),

(3, 4),

(3, 8),

(4, 5),

(4, 6),

(5, 7);

insert into albums

values(1, 'Confessions on dance floor', '20051111'),

(2, 'Love?', '20110502'),

(3, 'On the 6', '19990420'),

(4, 'Chosen', '20171207'),

(5, 'The Eminem show', '20100101'),

(6, 'The Lion King: The gift', '20190719'),

(7, 'Lound', '20100101'),

(8, 'Metallica', '19910101'),

(9, 'Kids see ghosts', '20180608'),

(10, 'Happier than ever', '20210730');

insert into artistalbum

values(1, 1),

(2, 2),

(2, 3),

(3, 4),

(4, 5),

(5, 6),

(6, 7),

(7, 8),

(8, 9),

(9, 10);

insert into tracks

values(1, 'Hung Up', '00:05:38', 1),

(2, 'Jump', '00:03:47', 1),

(3, 'Invading My Mind', '00:03:21', 2),

(4, 'If you had my love', '00:04:26', 3),

(5, 'Beggin', '3:31', 4),

(6, 'White America', '00:05:24', 5),

(7, 'Find your way bach', '00:02:42', 6),

(8, 'Fading', '00:3:19', 7),

(9, 'Sad but true', '00:6:55', 8),

(10, 'Feel the love', '00:02:45', 9),

(11, 'My future', '00:03:30', 10),

(12, 'Everybody dies', '00:03:26', 10),

(13, 'Oxytocin', '00:03:30', 10),

(14, 'My power', '00:04:19', 6),

(15, 'Recovery', '00:02:56', 4);

insert into collections

values(1, 'Collection#1', '20170101'),

(2, 'Collection#2', '20170601'),

(3, 'Collection#3', '20180101'),

(4, 'Collection#4', '20180601'),

(5, 'Collection#5', '20190101'),

(6, 'Collection#6', '20190601'),

(7, 'Collection#7', '20200101'),

(8, 'Collection#8', '20200601'),

(9, 'Collection#9', '20210801');

insert into CollectionTracks

values(1, 2),

(1, 4),

(1, 8),

(2, 3),

(2, 6),

(2, 9),

(3, 5),

(3, 1),

(3, 8),

(4, 15),

(4, 3),

(4, 6),

(5, 10),

(5, 2),

(5, 9),

(6, 8),

(6, 3),

(6, 4),

(7, 7),

(7, 8),

(7, 9),

(8, 2),

(8, 4),

(8, 7),

(9, 11),

(9, 12),

(9, 13);

insert into artistalbum

Values(6, 5)