Table for artist

CREATE TABLE artist (

name varchar(50) NOT NULL,

PRIMARY KEY (name)

);

Table for album

CREATE TABLE album (

name varchar(100) NOT NULL,

artist\_name varchar(50) NOT NULL,

release\_date date NOT NULL,

PRIMARY KEY (name),

KEY artist\_name\_idx (artist\_name),

CONSTRAINT artist\_name FOREIGN KEY (artist\_name) REFERENCES artist (name)

);

Table for song

CREATE TABLE song (

song\_name varchar(100) NOT NULL,

artist varchar(50) NOT NULL,

release\_date date DEFAULT NULL,

album\_name varchar(100) DEFAULT NULL,

PRIMARY KEY (song\_name,artist),

KEY artist\_idx (artist),

CONSTRAINT artist FOREIGN KEY (artist) REFERENCES artist (name)

);

Table for genre

CREATE TABLE genre (

genre varchar(20) NOT NULL,

PRIMARY KEY (genre)

);

Table for song and genre

CREATE TABLE song\_genre (

song\_name varchar(50) DEFAULT NULL,

genre varchar(20) DEFAULT NULL,

KEY song\_name\_idx (song\_name),

KEY genre\_idx (genre),

CONSTRAINT genre FOREIGN KEY (genre) REFERENCES genre (genre),

CONSTRAINT song\_name FOREIGN KEY (song\_name) REFERENCES song (song\_name)

);

Table for user

CREATE TABLE user (

user\_name varchar(100) NOT NULL,

PRIMARY KEY (user\_name)

);

Table for playlist

CREATE TABLE play\_list (

list\_name varchar(100) NOT NULL,

song\_name varchar(100) NOT NULL,

user\_name varchar(50) NOT NULL,

create\_date date NOT NULL,

PRIMARY KEY (list\_name,user\_name),

KEY user\_name\_idx (user\_name),

CONSTRAINT create\_user FOREIGN KEY (user\_name) REFERENCES user (user\_name)

);

Table for rating

CREATE TABLE rating (

user\_name varchar(50) NOT NULL,

name varchar(100) NOT NULL,

rating int NOT NULL,

rating\_date date NOT NULL,

KEY user\_name\_idx (user\_name),

CONSTRAINT user\_name FOREIGN KEY (user\_name) REFERENCES user (user\_name)

);

Query

1

select genre, count(\*) as number\_of\_songs

from song\_genre

group by genre

order by number\_of\_songs desc

limit 3;

2

select distinct artist\_name

from album

where artist\_name in (

select distinct artist

from song

where album\_name = null);

3

select name as album\_name, avg(rating) as average\_user\_rating

from rating

where name in (select name

from album

) and rating\_date between "1990-01-01" and "1999-12-31"

group by name

order by average\_user\_rating desc, name asc

limit 10;

4

select count(song\_genre.song\_name) as number\_of\_song\_ratings , song\_genre.genre as genre\_name

from song\_genre

right join rating on song\_genre.song\_name = rating.name

where rating is not null and rating.rating\_date between "1991-01-01" and "1995-12-31"

group by genre\_name

limit 3;

5

select play\_list.user\_name as username, play\_list.list\_name as playlist\_title, avg(rating.rating) as average\_song\_rating

from play\_list

right join rating on play\_list.song\_name = rating.name

group by list\_name

having average\_song\_rating > 4;

6

SELECT user\_name as username, count(\*) as number\_of\_ratings

from rating

where name in (select song\_name from song) or (select name from album)

group by username

order by number\_of\_ratings desc

limit 5;

7

select artist as artist\_name, count(\*) as number\_of\_songs

from song

where release\_date between "1990-01-01" and "2010-12-31"

group by artist

order by number\_of\_songs desc

limit 10;

8

select song\_name as song\_title, count(song\_name) as number\_of\_playlists

from play\_list

group by song\_name

order by number\_of\_playlists desc, song\_title asc

limit 10;

9

select song\_name as song\_title , genre as Genre\_name, count as number\_of\_ratings

from

(select \*, row\_number() over(partition by genre order by count desc, song\_name asc) as rnk

from

(select song\_name, genre, count(rating) as count

from song\_genre

join rating on song\_name = rating.name

group by genre, song\_name) a) s

where rnk between 1 and 20;

10

select distinct artist

from song

where artist not in (select artist from song where release\_date > "1993-12-31");