Al Alamein International University

Faculty of Computer Science & Engineering

CSE221 Database Systems

voice club management system

Team Members:

- **nour basem**
- **salah ezzat**
- **mariam tahoon**

❖ Application Description:

The Voice Club Management System is a database system designed to facilitate the management of a voice club. The system will allow users to create and manage voice club events, keep track of member information, and maintain a database of song lyrics and music files. The system will also provide users with the ability to search for songs and create playlists.

❖ Business Rules:

1. Each member must have a unique ID.

2. Only registered members can participate in club events.

3. Club events must be approved by the club's administrators.

4. Members can only vote on songs they have heard.

5. Song requests must be made in advance of club events.

6. The club's administrators have the final say on all decisions.

❖Potential Queries:

1. What are the top 10 most requested songs?

2. Who are the members with the most song requests?

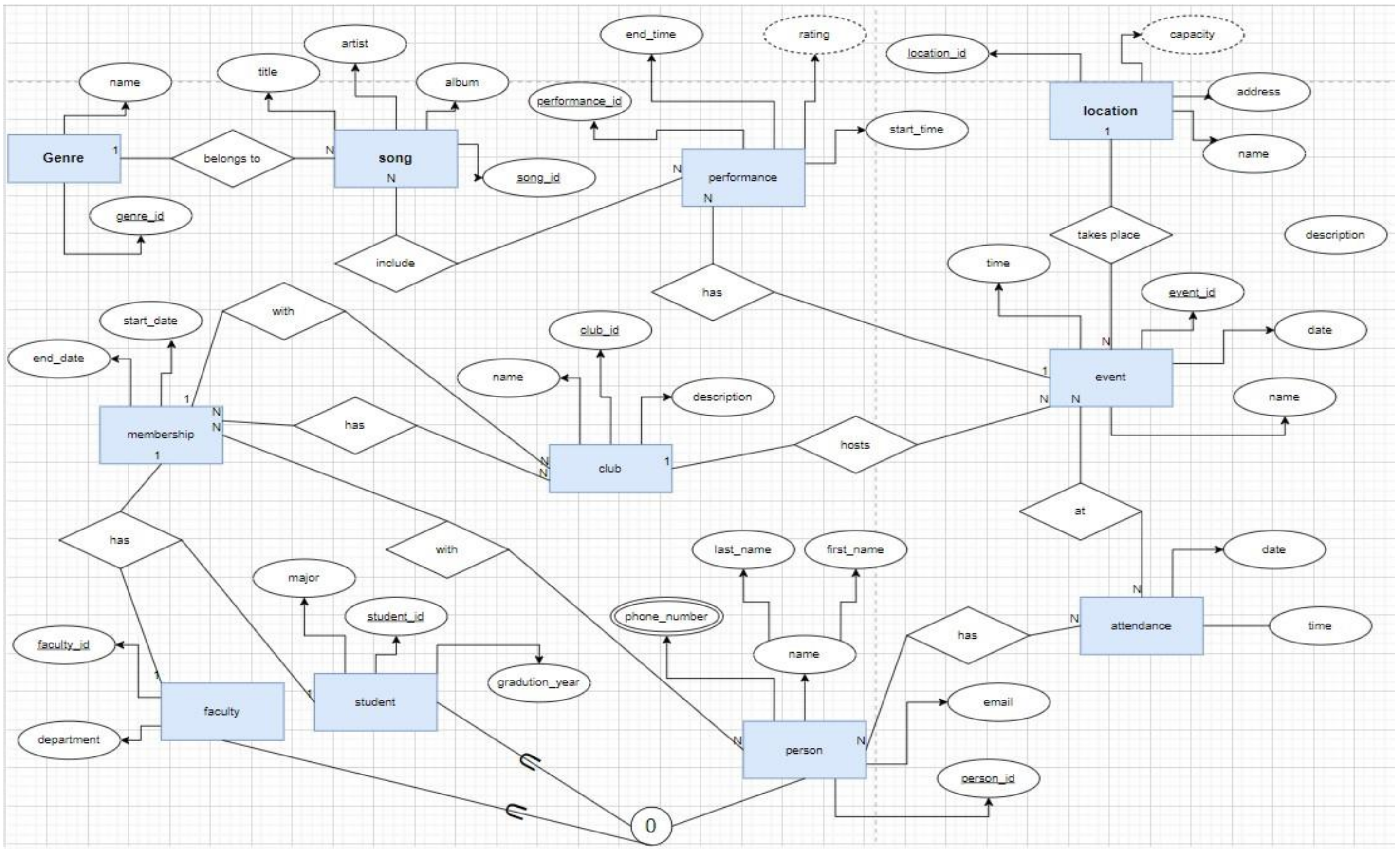
3. What are the lyrics of a specific song?

4. How many members attended a particular club event?

5. What is the average rating for a particular song?

6. Which members have not attended any club events

❖ EERD:



❖Relational Schema:

- genre (genre_id , name)
- song (song_id , album, artist , title , genre_id)

foreign key song (genre_id) references genre (genre_id)

- performance (performance_id , end_time , start_time , event_id)

foreign key performance (event_id) references genre event (event_id)

- location (location_id , address, name)
- event (event_id , time , date , description , location_id , club_id)

foreign key event (location_id) references location (location_id)

foreign key event (club_id) references club (club_id)

- attendance (date , time)
- person(person_id , email, last_name, first_name)

• person_phoneNum(phone_number , person_id)

foreign key person_phoneNum (person_id) references person (person_id)

• student (student_id , major , gradution_year)

• faculty(faculty_id , department)

• membership(end_time, start_time , faculty_id , student_id)

foreign key membership (faculty_id) references faculty (faculty_id)

foreign key membership (student_id) references student (student_id)

• club (club_id , name, description, end_date ,start_date)

• include(song_id , performance_id)

foreign key include (song_id) references song (song_id)

foreign key include (performance_id) references performance (performance_id)

• at(event_id , date, time)

foreign key at (event_id) references event (event_id)

• has(end_time, start_time , club_id)

foreign key has (club_id) references club (club_id)

• has(time , date , person_id)

foreign key has (person_id) references person (person_id)

• with(end_time, start_time , person_id)

foreign key with (person_id) references person (person_id)

Genre: represents the genre of the song. It has attributes genre_id and name.

Department: represents a department in a faculty. It has attributes dept_id and name.

Faculty: represents a faculty of a university. It has attributes faculty_id, name, and dept_id.

Major: represents a student's major. It has attributes major_id and name.

Student: represents a student in the university. It has attributes student_id, first_name, last_name, major_id, and grad_year.

Club: represents a club in the university. It has attributes club_id, name, and faculty_id.

Performance: represents a performance by a singer or a group in a club. It has attributes performance_id, name, date, start_time, end_time, location_id, and club_id.

Artist: represents an artist who can sing or perform. It has attributes artist_id and name.

Song: represents a song that is sung by an artist. It has attributes song_id, name, and artist_id. Album: represents an album that contains multiple songs. It has attributes album_id, name, and artist_id.

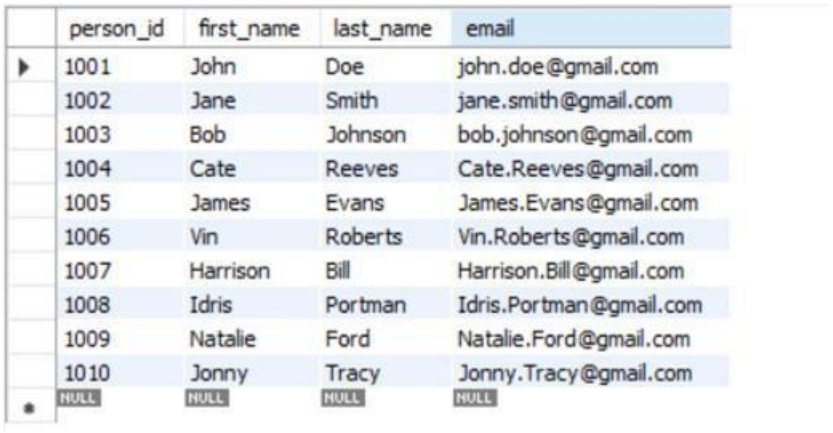
Singer: represents a singer who performs in a performance. It has attributes singer_id and name. Sang: represents a singer's performance in a performance. It has attributes performance_id and singer_id.

Event: represents an event that is held in the university. It has attributes event_id, name, description, date, time, location_id, and capacity.

Attendance: represents a student's attendance in an event. It has attributes event_id and student_id

Person

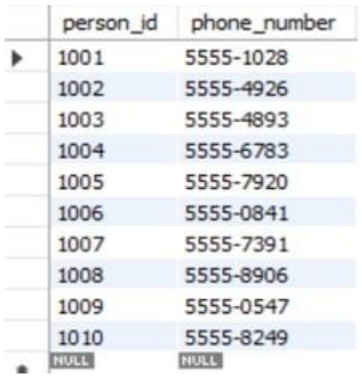
```
create database dataa;
use dataa;

CREATE TABLE Person (
person_id INT NOT NULL PRIMARY KEY,
first_name VARCHAR(50) NOT NULL,
last_name VARCHAR(50) NOT NULL,
email VARCHAR(100)  UNIQUE NOT NULL
);

INSERT INTO Person (person_id, first_name, last_name, email)VALUES
(1001, 'John', 'Doe', 'john.doe@gmail.com'),
(1002, 'Jane', 'Smith', 'jane.smith@gmail.com'),
(1003, 'Bob', 'Johnson', 'bob.johnson@gmail.com'),
(1004, 'Cate', 'Reeves', 'Cate.Reeves@gmail.com'),
(1005, 'James', 'Evans', 'James.Evans@gmail.com'),
(1006, 'Vin', 'Roberts', 'Vin.Roberts@gmail.com'),
(1007, 'Harrison', 'Bill', 'Harrison.Bill@gmail.com'),
(1008, 'Idris', 'Portman', 'Idris.Portman@gmail.com'),
(1009, 'Natalie', 'Ford', 'Natalie.Ford@gmail.com'),
(1010, 'Jonny', 'Tracy', 'Jonny.Tracy@gmail.com');

 select * from Person;
```



person_id	first_name	last_name	email
1001	John	Doe	john.doe@gmail.com
1002	Jane	Smith	jane.smith@gmail.com
1003	Bob	Johnson	bob.johnson@gmail.com
1004	Cate	Reeves	Cate.Reeves@gmail.com
1005	James	Evans	James.Evans@gmail.com
1006	Vin	Roberts	Vin.Roberts@gmail.com
1007	Harrison	Bill	Harrison.Bill@gmail.com
1008	Idris	Portman	Idris.Portman@gmail.com
1009	Natalie	Ford	Natalie.Ford@gmail.com
1010	Jonny	Tracy	Jonny.Tracy@gmail.com
NULL	NULL	NULL	NULL

Person_PhoneNum

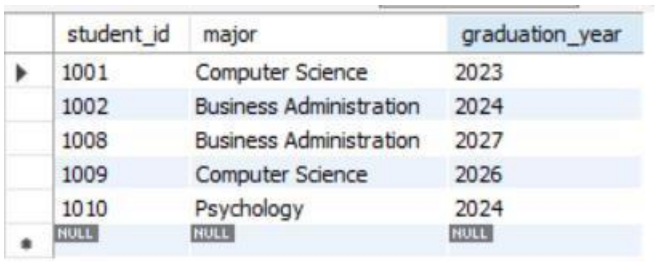
```
 CREATE TABLE Person_PhoneNum (
person_id INT NOT NULL PRIMARY KEY,
phone_number varchar(52) NOT NULL,
FOREIGN KEY (person_id) REFERENCES Person(person_id)
);

INSERT INTO Person_PhoneNum (person_id, phone_number)VALUES
(1001, '5555-1028' ),
(1002, '5555-4926' ),
(1003, '5555-4893' ),
(1004, '5555-6783' ),
(1005, '5555-7920' ),
(1006, '5555-0841' ),
(1007, '5555-7391' ),
(1008, '5555-8906' ),
(1009, '5555-0547' ),
(1010, '5555-8249' );

 select * from Person_PhoneNum;
```



person_id	phone_number
1001	5555-1028
1002	5555-4926
1003	5555-4893
1004	5555-6783
1005	5555-7920
1006	5555-0841
1007	5555-7391
1008	5555-8906
1009	5555-0547
1010	5555-8249
NULL	NULL

Student

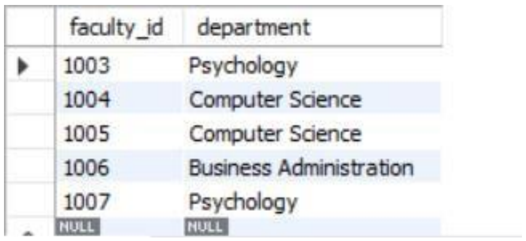
```
 CREATE TABLE Student (
student_id INT NOT NULL PRIMARY KEY,
major VARCHAR(50) NOT NULL,
graduation_year INT NOT NULL,
FOREIGN KEY (student_id) REFERENCES Person(person_id)
);

INSERT INTO Student (student_id, major, graduation_year)
VALUES
(1001, 'Computer Science', 2023),
(1002, 'Business Administration', 2024),
(1008, 'Business Administration', 2027),
(1009, 'Computer Science', 2026),
(1010, 'Psychology', 2024);

select * from Student;

UPDATE Person SET email = 'john.doe@student.gmail.com' WHERE person_id = 1001;
UPDATE Person SET email = 'jane.smith@student.gmail.com' WHERE person_id = 1002;
UPDATE Person SET email = 'Idris.Portman@student.gmail.com' WHERE person_id = 1008;
UPDATE Person SET email = 'Natalie.Ford@student.gmail.com' WHERE person_id = 1009;
UPDATE Person SET email = 'Jonny.Tracy@student.gmail.com' WHERE person_id = 1010;


select * from Student;
```



student_id	major	graduation_year
1001	Computer Science	2023
1002	Business Administration	2024
1008	Business Administration	2027
1009	Computer Science	2026
1010	Psychology	2024
NULL	NULL	NULL

Faculty

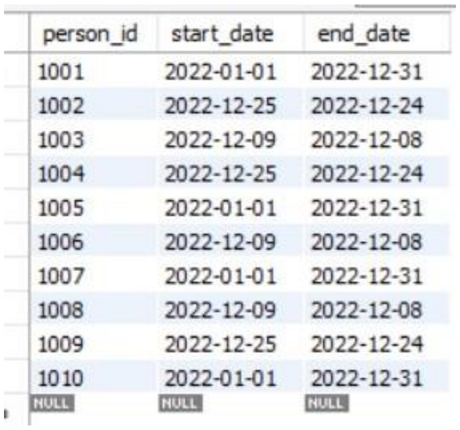
```
CREATE TABLE Faculty (
faculty_id INT NOT NULL PRIMARY KEY,
department VARCHAR(50) NOT NULL,
FOREIGN KEY (faculty_id) REFERENCES Person(person_id)
);

INSERT INTO Faculty (faculty_id, department)
VALUES
(1003, 'Psychology'),
(1004, 'Computer Science'),
(1005, 'Computer Science'),
(1006, 'Business Administration'),
(1007, 'Psychology');

UPDATE Person SET email = 'bob.johnson@faculty.gmail.com' WHERE person_id = 1003;
UPDATE Person SET email = 'Cate.Reeves@faculty.gmail.com' WHERE person_id = 1004 ;
UPDATE Person SET email = 'James.Evans@faculty.gmail.com' WHERE person_id = 1005 ;
UPDATE Person SET email = 'Vin.Roberts@faculty.gmail.com' WHERE person_id = 1006;
UPDATE Person SET email = 'Harrison.Bill@faculty.gmail.com' WHERE person_id = 1007;

select * from faculty;
```



faculty_id	department
1003	Psychology
1004	Computer Science
1005	Computer Science
1006	Business Administration
1007	Psychology
NULL	NULL

Membership

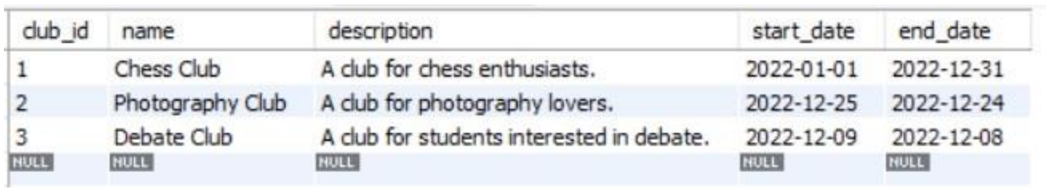
```
CREATE TABLE Membership (
person_id INT NOT NULL,
start_date DATE NOT NULL,
end_date DATE NOT NULL,
PRIMARY KEY (person_id),
FOREIGN KEY (person_id) REFERENCES Person(person_id)
);

INSERT INTO Membership (person_id , start_date, end_date)
VALUES
(1001, '2022-01-01', '2022-12-31'),
(1002, '2022-12-25', '2022-12-24'),
(1003, '2022-12-09', '2022-12-08'),
(1004, '2022-12-25', '2022-12-24'),
(1005, '2022-01-01', '2022-12-31'),
(1006, '2022-12-09', '2022-12-08'),
(1007, '2022-01-01', '2022-12-31'),
(1008, '2022-12-09', '2022-12-08'),
(1009, '2022-12-25', '2022-12-24'),
(1010, '2022-01-01', '2022-12-31');

select * from Membership;
```



person_id	start_date	end_date
1001	2022-01-01	2022-12-31
1002	2022-12-25	2022-12-24
1003	2022-12-09	2022-12-08
1004	2022-12-25	2022-12-24
1005	2022-01-01	2022-12-31
1006	2022-12-09	2022-12-08
1007	2022-01-01	2022-12-31
1008	2022-12-09	2022-12-08
1009	2022-12-25	2022-12-24
1010	2022-01-01	2022-12-31
NULL	NULL	NULL

Club

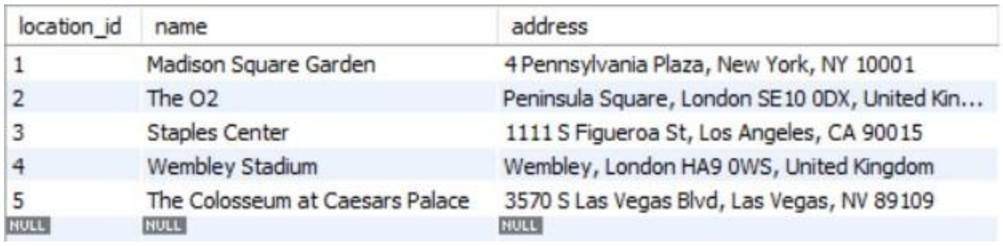
```
CREATE TABLE Club (
club_id INT NOT NULL PRIMARY KEY,
name VARCHAR(50) NOT NULL,
description VARCHAR(100) NOT NULL,
start_date DATE NOT NULL,
end_date DATE NOT NULL
);

INSERT INTO Club (club_id, name, description,start_date,end_date)
VALUES
(1, 'Chess Club', 'A club for chess enthusiasts.','2022-01-01', '2022-12-31'),
(2, 'Photography Club', 'A club for photography lovers.','2022-12-25', '2022-12-24'),
(3, 'Debate Club', 'A club for students interested in debate.','2022-12-09', '2022-12-08');

 select * from club;
```



club_id	name	description	start_date	end_date
1	Chess Club	A club for chess enthusiasts.	2022-01-01	2022-12-31
2	Photography Club	A club for photography lovers.	2022-12-25	2022-12-24
3	Debate Club	A club for students interested in debate.	2022-12-09	2022-12-08
NULL	NULL	NULL	NULL	NULL

Location

```
CREATE TABLE Location (
location_id INT NOT NULL PRIMARY KEY,
name VARCHAR(50) NOT NULL,
address VARCHAR(100) NOT NULL
);

INSERT INTO Location (location_id, name, address)
VALUES
(1, 'Madison Square Garden', '4 Pennsylvania Plaza, New York, NY 10001'),
(2, 'The O2', 'Peninsula Square, London SE10 0DX, United Kingdom'),
(3, 'Staples Center', '1111 S Figueroa St, Los Angeles, CA 90015'),
(4, 'Wembley Stadium', 'Wembley, London HA9 0WS, United Kingdom'),
(5, 'The Colosseum at Caesars Palace', '3570 S Las Vegas Blvd, Las Vegas, NV 89109');

select * from location;
```

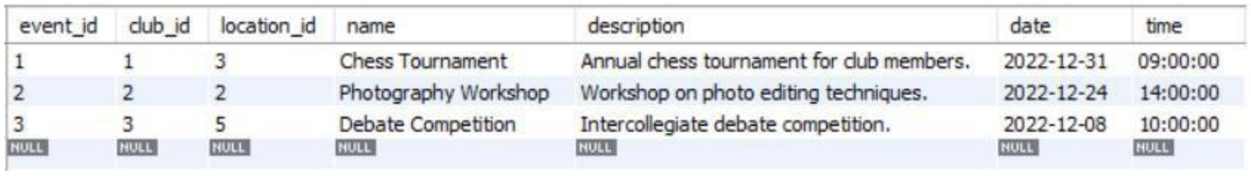


location_id	name	address
1	Madison Square Garden	4 Pennsylvania Plaza, New York, NY 10001
2	The O2	Peninsula Square, London SE10 0DX, United Kin...
3	Staples Center	1111 S Figueroa St, Los Angeles, CA 90015
4	Wembley Stadium	Wembley, London HA9 0WS, United Kingdom
5	The Colosseum at Caesars Palace	3570 S Las Vegas Blvd, Las Vegas, NV 89109
NULL	NULL	NULL

Event

```
 CREATE TABLE Event (
event_id INT NOT NULL PRIMARY KEY,
club_id INT NOT NULL,
location_id INT NOT NULL,
name VARCHAR(50) NOT NULL,
description VARCHAR(100) NOT NULL,
date DATE NOT NULL,
time TIME NOT NULL,
FOREIGN KEY (club_id) REFERENCES Club(club_id),
FOREIGN KEY (location_id) REFERENCES location(location_id)
);

INSERT INTO Event (event_id, club_id,location_id, name, description, date, time)
VALUES
(1, 1, 3,'Chess Tournament', 'Annual chess tournament for club members.', '2022-12-31',
'09:00:00'),
(2, 2, 2, 'Photography Workshop', 'Workshop on photo editing techniques.', '2022-12-24',
'14:00:00'),
(3, 3, 5,'Debate Competition', 'Intercollegiate debate competition.', '2022-12-8', '10:00:00');
```

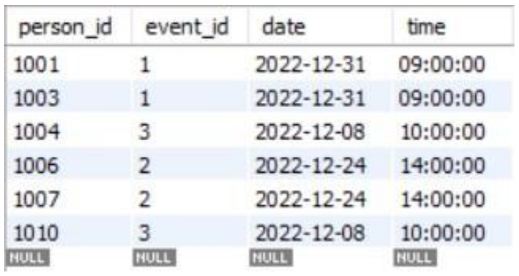
select * from event;



event_id	club_id	location_id	name	description	date	time
1	1	3	Chess Tournament	Annual chess tournament for club members.	2022-12-31	09:00:00
2	2	2	Photography Workshop	Workshop on photo editing techniques.	2022-12-24	14:00:00
3	3	5	Debate Competition	Intercollegiate debate competition.	2022-12-08	10:00:00
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Attendance

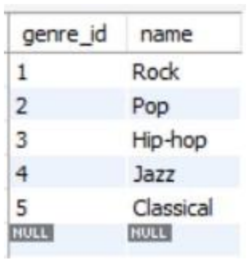
```
CREATE TABLE Attendance (
person_id INT NOT NULL,
event_id INT NOT NULL,
date DATE NOT NULL,
time TIME NOT NULL,
PRIMARY KEY (person_id, event_id),
FOREIGN KEY (person_id) REFERENCES Person(person_id),
FOREIGN KEY (event_id) REFERENCES Event(event_id)
);

INSERT INTO Attendance (person_id, event_id, date, time)
VALUES
(1001, 1, '2022-12-31', '09:00:00'),
(1003, 1, '2022-12-31', '09:00:00'),
(1004, 3, '2022-12-8', '10:00:00'),
(1006, 2, '2022-12-24', '14:00:00'),
(1007, 2, '2022-12-24', '14:00:00'),
(1010, 3, '2022-12-8', '10:00:00');

select * from Attendance;
```



person_id	event_id	date	time
1001	1	2022-12-31	09:00:00
1003	1	2022-12-31	09:00:00
1004	3	2022-12-08	10:00:00
1006	2	2022-12-24	14:00:00
1007	2	2022-12-24	14:00:00
1010	3	2022-12-08	10:00:00
NULL	NULL	NULL	NULL

Genre

```
CREATE TABLE Genre (
genre_id INT NOT NULL PRIMARY KEY,
name VARCHAR(50) NOT NULL
);

INSERT INTO Genre (genre_id, name)
VALUES
(1, 'Rock'),
(2, 'Pop'),
(3, 'Hip-hop'),
(4, 'Jazz'),
(5, 'Classical');

select * from Genre;
```

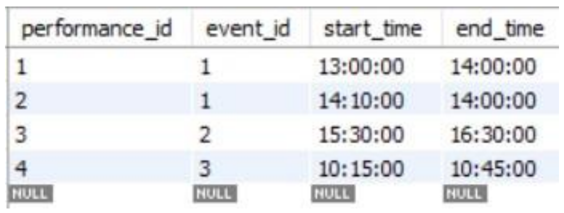


genre_id	name
1	Rock
2	Pop
3	Hip-hop
4	Jazz
5	Classical
NULL	NULL

Performance

```
CREATE TABLE Performance (
performance_id INT NOT NULL PRIMARY KEY,
event_id INT NOT NULL,
start_time TIME NOT NULL,
end_time TIME NOT NULL,
FOREIGN KEY (event_id) REFERENCES Event(event_id)
);
INSERT INTO Performance (performance_id, event_id, start_time, end_time)
VALUES
(1, 1, '13:00:00', '14:00:00'),
(2, 1, '14:10:00', '14:00:00'),
(3, 2, '15:30:00', '16:30:00'),
(4, 3, '10:15:00', '10:45:00');
```

select * from Performance;



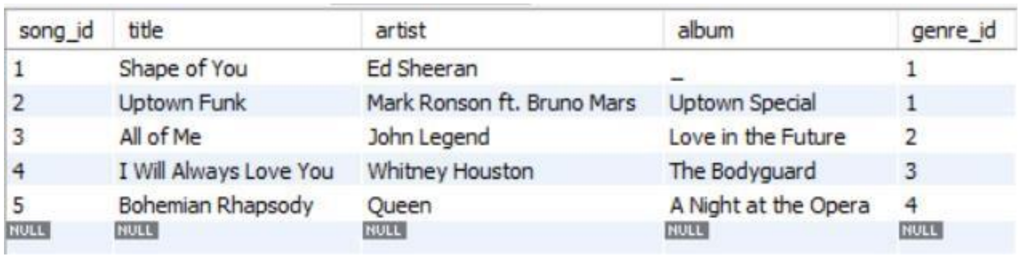
performance_id	event_id	start_time	end_time
1	1	13:00:00	14:00:00
2	1	14:10:00	14:00:00
3	2	15:30:00	16:30:00
4	3	10:15:00	10:45:00
NULL	NULL	NULL	NULL

Song

```
CREATE TABLE Song (
song_id INT NOT NULL PRIMARY KEY,
title VARCHAR(50) NOT NULL,
artist VARCHAR(50) NOT NULL,
album VARCHAR(50) NOT NULL,
genre_id INT NOT NULL,
FOREIGN KEY (genre_id) REFERENCES Genre(genre_id)
);
```

```
INSERT INTO Song (song_id, title, artist, album, genre_id)
VALUES
(1, 'Shape of You', 'Ed Sheeran', '_', 1),
(2, 'Uptown Funk', 'Mark Ronson ft. Bruno Mars', 'Uptown Special', 1),
(3, 'All of Me', 'John Legend', 'Love in the Future', 2),
(4, 'I Will Always Love You', 'Whitney Houston', 'The Bodyguard', 3),
(5, 'Bohemian Rhapsody', 'Queen', 'A Night at the Opera', 4);
```

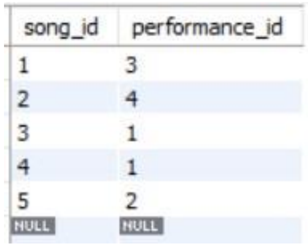
select * from song;



song_id	title	artist	album	genre_id
1	Shape of You	Ed Sheeran	_	1
2	Uptown Funk	Mark Ronson ft. Bruno Mars	Uptown Special	1
3	All of Me	John Legend	Love in the Future	2
4	I Will Always Love You	Whitney Houston	The Bodyguard	3
5	Bohemian Rhapsody	Queen	A Night at the Opera	4
NULL	NULL	NULL	NULL	NULL

include

```
CREATE TABLE include (
song_id INT NOT NULL ,
performance_id INT NOT NULL,
PRIMARY KEY (song_id,performance_id),
FOREIGN KEY (song_id) REFERENCES song(song_id),
FOREIGN KEY (performance_id) REFERENCES performance(performance_id)
);


INSERT INTO include (song_id, performance_id)
VALUES
(1, 3),
(2, 4 ),
(3, 1),
(4, 1),
(5, 2);

select * from include;
```

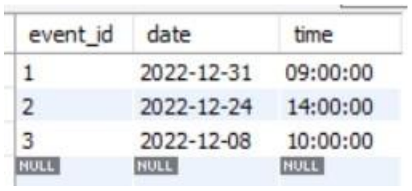


song_id	performance_id
1	3
2	4
3	1
4	1
5	2
NULL	NULL

at

```
CREATE TABLE at (
event_id INT NOT NULL PRIMARY KEY ,
date DATE NOT NULL,
time TIME NOT NULL,
FOREIGN KEY (event_id) REFERENCES event(event_id)
);

INSERT INTO at (event_id, date, time)
VALUES
(1, '2022-12-31', '09:00:00'),
(2,'2022-12-24', '14:00:00' ),
(3, '2022-12-8', '10:00:00');
```

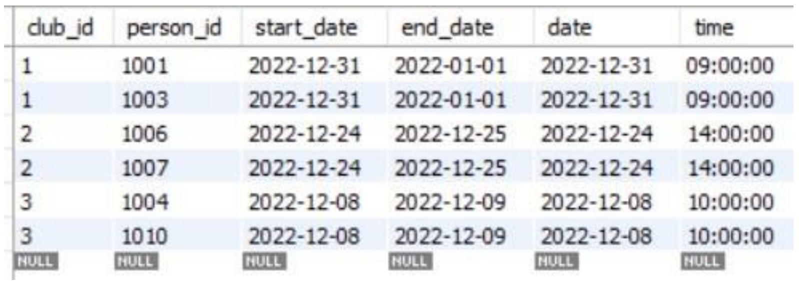
select * from at;



event_id	date	time
1	2022-12-31	09:00:00
2	2022-12-24	14:00:00
3	2022-12-08	10:00:00
NULL	NULL	NULL

has

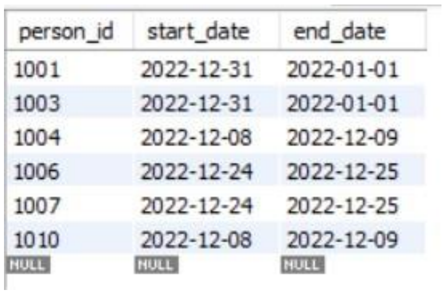
```
CREATE TABLE has (
club_id INT NOT NULL,
person_id INT NOT NULL,
start_date DATE NOT NULL,
end_date DATE NOT NULL,
date DATE NOT NULL,
time TIME NOT NULL,
PRIMARY KEY (club_id , person_id),
FOREIGN KEY (club_id) REFERENCES club(club_id),
FOREIGN KEY (person_id) REFERENCES person(person_id)
);

INSERT INTO has (person_id , club_id  ,date, time, end_date ,start_date)
VALUES
(1001,1, '2022-12-31', '09:00:00', '2022-01-01', '2022-12-31'),
(1003,1, '2022-12-31', '09:00:00', '2022-01-01', '2022-12-31'),
(1004,3, '2022-12-8', '10:00:00','2022-12-09', '2022-12-08'),
(1006,2,'2022-12-24',  '14:00:00'  ,'2022-12-25',  '2022-12-24'),
(1007,2,'2022-12-24',  '14:00:00'  ,'2022-12-25',  '2022-12-24'),
(1010,3, '2022-12-8', '10:00:00','2022-12-09', '2022-12-08');

select * from has;
```



club_id	person_id	start_date	end_date	date	time
1	1001	2022-12-31	2022-01-01	2022-12-31	09:00:00
1	1003	2022-12-31	2022-01-01	2022-12-31	09:00:00
2	1006	2022-12-24	2022-12-25	2022-12-24	14:00:00
2	1007	2022-12-24	2022-12-25	2022-12-24	14:00:00
3	1004	2022-12-08	2022-12-09	2022-12-08	10:00:00
3	1010	2022-12-08	2022-12-09	2022-12-08	10:00:00
NULL	NULL	NULL	NULL	NULL	NULL

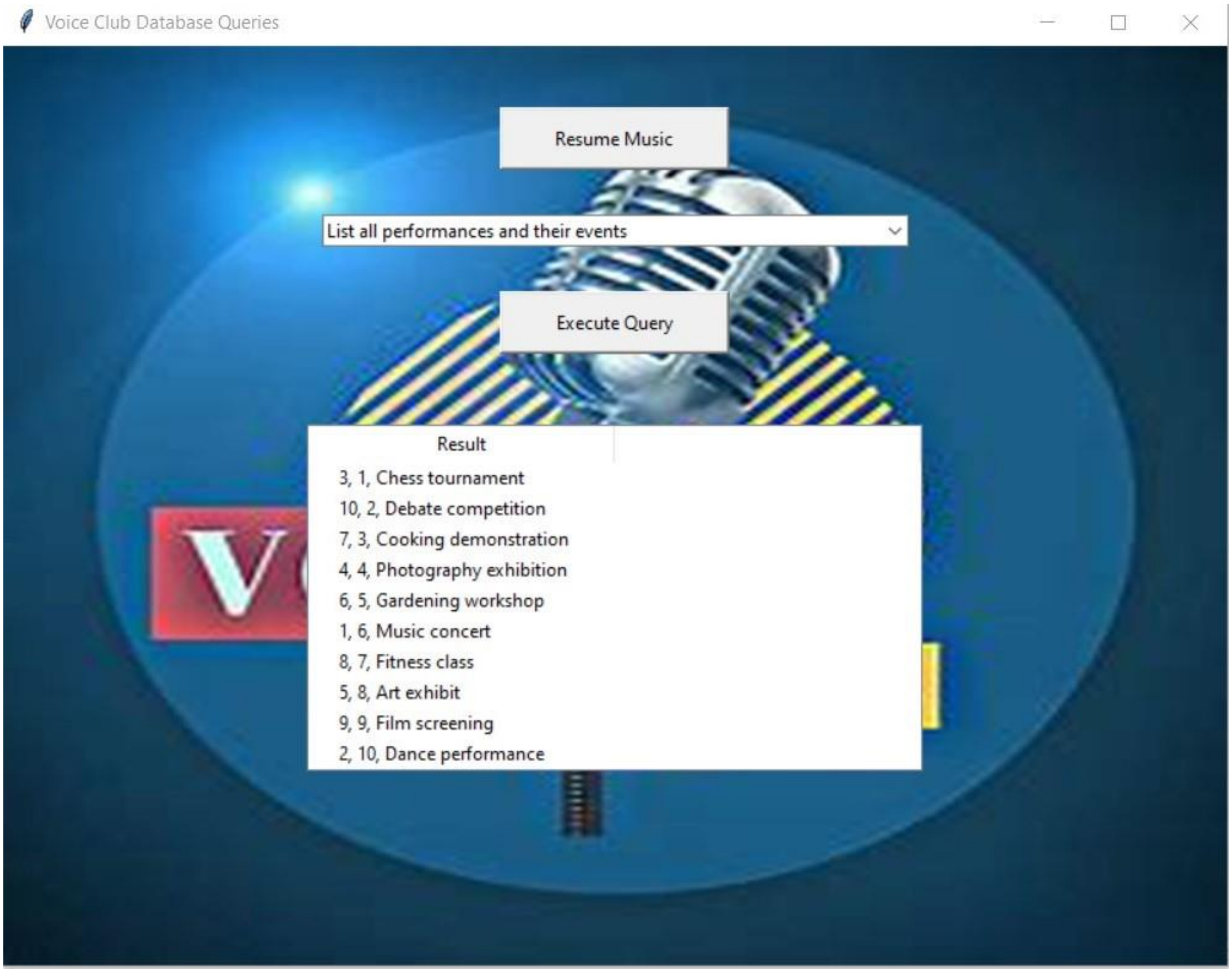
withh

```
CREATE TABLE withh (
person_id INT NOT NULL primary key,
start_date DATE NOT NULL,
end_date DATE NOT NULL,
FOREIGN KEY (person_id) REFERENCES person(person_id)
);

INSERT INTO withh (person_id , end_date ,start_date)
VALUES
(1001, '2022-01-01', '2022-12-31'),
(1003, '2022-01-01', '2022-12-31'),
(1004,'2022-12-09',  '2022-12-08'),
(1006,'2022-12-25',  '2022-12-24'),
(1007,'2022-12-25',  '2022-12-24'),
(1010,'2022-12-09',  '2022-12-08');

select * from withh;
```



person_id	start_date	end_date
1001	2022-12-31	2022-01-01
1003	2022-12-31	2022-01-01
1004	2022-12-08	2022-12-09
1006	2022-12-24	2022-12-25
1007	2022-12-24	2022-12-25
1010	2022-12-08	2022-12-09
NULL	NULL	NULL

interface :


Voice Club Database Queries
Resume Music
List all performances and their events
Execute Query
Result

```
3, 1, Chess tournament
10, 2, Debate competition
7, 3, Cooking demonstration
4, 4, Photography exhibition
6, 5, Gardening workshop
1, 6, Music concert
8, 7, Fitness class
5, 8, Art exhibit
9, 9, Film screening
2, 10, Dance performance
```

This program is a graphical interface for a voice club database that allows the user to select and execute predefined queries. The interface is built using the tkinter module and includes a Combobox widget to select the query, a Button widget to execute it, and a Listbox widget to display the results. Additionally, the interface includes a button to pause and resume the background music, which is played using the pygame module.

The program connects to a MySQL database using the mysql.connector module and defines a list of queries to be executed. Each query is represented as a tuple containing a description and the SQL code to execute. The execute_query() function takes a query as input, executes it using a database cursor, and returns the results as a list of strings. The display_results() function takes the results as input and updates the Listbox widget to display them.

The program also includes code to download and display a wallpaper image using the requests and PIL modules. The wallpaper is displayed as a Label widget using the tkinter interface.

Finally, the program includes code to play and pause background music using the pygame.mixer module. The music file is loaded using pygame.mixer.music.load() and played using pygame.mixer.music.play(). The toggle_music() function is used to pause and resume the music playback, and the music_paused variable is used to keep track of the current playback state.

Overall, the program provides a user-friendly interface for executing predefined queries on a voice club database, with additional features such as background music and wallpaper.