

Оглавление

1. Описание предметной области.	1
3. ER-модель базы данных.	2
4. Описание схемы базы данных на языке SQL.	3
5. Описание используемых запросов на языке SQL.	5
6. Руководство пользователя по работе с приложением.	8

1. Описание предметной области.

Филармония.

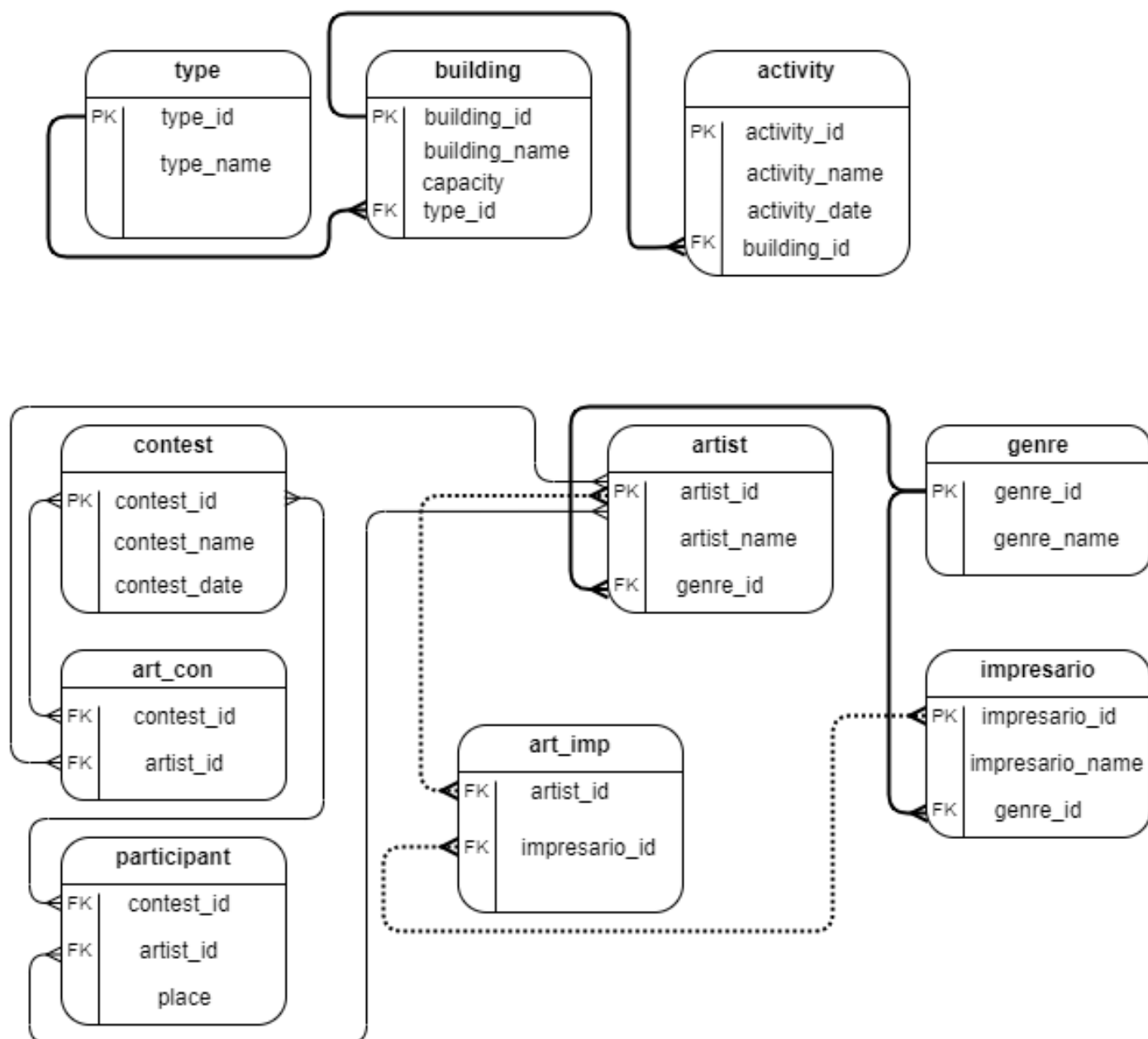
Инфраструктура городской филармонии представлена культурными сооружениями различного типа: театры, концертные площадки, эстрады, дворцы культуры и т.д.

Каждая из категорий культурных сооружений обладает атрибутами, специфичными только для нее: театр характеризуется вместимостью, кинотеатр - размером экрана. Артисты под руководством импресарио выступают в различных жанрах, при этом один и тот же артист может выступать в нескольких жанрах, и может работать с несколькими импресарио. Организаторы концертных мероприятий проводят выступления, концерты, конкурсы в культурных сооружениях города, организуя участие в нем артистов.

По результатам участия артистов в конкурсах производится награждение. Система должна эффективно выдавать ответы на следующие запросы:

1. Получить перечень культурных сооружений указанного типа в целом или удовлетворяющих заданным характеристикам (например, залы, вмещающие не менее заданного числа зрителей).
2. Получить список артистов, выступающих в некотором жанре.
3. Получить список артистов, работающих с некоторым импресарио.
4. Получить список артистов, работающих более чем с одним импресарио с их указанием.
5. Получить список импресарио указанного артиста.
6. Получить перечень концертных мероприятий, проведенных в течение заданного периода времени в целом либо указанным организатором.
7. Получить список призеров указанного конкурса.
8. Получить перечень концертных мероприятий, проведенных в указанном культурном сооружении.
9. Получить список импресарио определенного жанра.
10. Получить список артистов, участвовавших в конкурсах в течение определенного периода времени.
11. Получить список организаторов культурных мероприятий и число проведенных ими концертов в течение определенного периода времени.
12. Получить перечень культурных сооружений, а также даты проведения на них культурных мероприятий в течение определенного периода времени

3. ER-модель базы данных.



4. Описание схемы базы данных на языке SQL.

1. Создание таблицы **genre** и заполнение значений:

```
CREATE TABLE genre(  
    genre_id INTEGER PRIMARY KEY,  
    genre_name VARCHAR(20));  
  
INSERT INTO genre(genre_name)  
VALUES ('Драма'), ('Комедия'), ('Трагедия'), ('Мюзикл'), ('Пьеса');
```

2. Создание таблицы **artist** и заполнение значений:

```
CREATE TABLE artist(  
    artist_id INTEGER PRIMARY KEY,  
    artist_name VARCHAR(20),  
    genre_id INT NOT NULL,  
    FOREIGN KEY (genre_id) REFERENCES genre (genre_id));  
  
INSERT INTO artist(artist_name, genre_id)  
VALUES ('Иванов И.И.', 1), ('Петров П.П.', 2), ('Ветров В.В.', 3), ('Семенов С.П.', 4),  
('Васильев К.К.', 5), ('Небов Н.И.', 1), ('Гладков И.И.', 2), ('Арбузов В.О.', 3), ('Зорина  
А.А.', 4), ('Малова У.К.', 5), ('Орова Э.С.', 1), ('Маринина У.П.', 2), ('Большакова Ф.Т.',  
3), ('Раневская Ф.Г.', 4), ('Жаров Я.Ф.', 5), ('Зудин И.Е.', 1), ('Понасенков Е.А.', 2),  
('Байден М.В.', 3), ('Горбачёв М.С.', 4), ('Набиуллина Э.С.', 5);
```

3. Создание таблицы **impresario** и заполнение значений:

```
CREATE TABLE impresario(  
    impresario_id INTEGER PRIMARY KEY,  
    impresario_name VARCHAR(20),  
    genre_id INT NOT NULL,  
    FOREIGN KEY (genre_id) REFERENCES genre (genre_id));  
  
INSERT INTO impresario(impresario_name, genre_id)  
VALUES ('Кошкин Ш.И.', 1), ('Шувалов Ф.П.', 2), ('Волков В.О.', 3), ('Медведев О.П.', 4),  
('Собакин Х.К.', 5), ('Зайцев А.И.', 1), ('Куручкин Г.И.', 2), ('Золотовицкий А.А.', 3),  
('Серебряков Н.А.', 4), ('Земцов Е.Е.', 5), ('Орлов А.С.', 1), ('Муравьев А.П.', 2),  
('Шувалов А.Т.', 3), ('Ранцова И.Е.', 4), ('Нигай А.Ф.', 5), ('Абрамова Е.Е.', 1), ('Позова  
К.А.', 2), ('Степина М.В.', 3), ('Лосева О.Л.', 4), ('Носов Ф.Я.', 5);
```

4. Создание вспомогательной таблицы **art_imp** и заполнение значений:

```
CREATE TABLE art_imp(  
    artist_id INT NOT NULL,  
    impresario_id INT NOT NULL,  
    FOREIGN KEY (artist_id) REFERENCES artist (artist_id),  
    FOREIGN KEY (impresario_id) REFERENCES impresario (impresario_id) );  
  
INSERT INTO art_imp(artist_id, impresario_id)
```

```
VALUES (1, 2), (1, 3), (1, 1), (2, 2), (2, 1), (3, 2), (3, 3), (4, 1), (5, 15), (5, 1), (6, 2), (6, 15), (6, 3), (7, 15), (8, 1), (8, 2), (8, 9), (9, 15), (10, 4), (10, 9), (11, 2), (12, 3), (13, 5), (14, 15), (15, 7), (16, 15), (18, 16), (19, 3), (20, 5), (20, 1);
```

5. Создание таблицы **type** и заполнение значений:

```
CREATE TABLE type(
type_id INTEGER PRIMARY KEY,
type_name VARCHAR(30) );
INSERT INTO type(type_name)
VALUES ('Концертный зал'), ('Зал церемоний'), ('Актный зал'),
('Царский зал'), ('Малая сцена'), ('Оркестровый зал'), ('Сцена театра мюзикла');
```

6. Создание таблицы **building** и заполнение значений:

```
CREATE TABLE building (
building_id INTEGER PRIMARY KEY,
building_name VARCHAR(30),
capacity INT,
type_id INT NOT NULL,
FOREIGN KEY (type_id) REFERENCES type (type_id));
INSERT INTO building(building_name, capacity, type_id)
VALUES ('Театр Наций', 1200, 4), ('Театр на Сretenке' , 250 , 5), ('Консерватория им. Чайковского', 430 , 1), ('Народный Центр', 800, 2), ('Дом Культуры' , 380 , 3), ('Консерватория им. Гнесина', 770 , 6), ('Театр Мюзикла', 1000 , 7), ('Театр Маяковского', 1500, 4), ('Театр Моссовета' , 170 , 5), ('Консерватория им. Магомаева', 430 , 1), ('Культурный Центр', 500, 2), ('Дом Творчества' , 180 , 3), ('Студия им. Моргенштерна', 666 , 6), ('Театр Музыки', 750 , 7), ('РокОркестра', '2022-07-01', 13), ('Песни о главном', '2021-10-12', 10), ('Лебединое Озеро', '2022-12-05', 6);
```

7. Создание таблицы **activity** и заполнение значений:

```
CREATE TABLE activity (
activity_id INTEGER PRIMARY KEY,
activity_name VARCHAR(30),
activity_date DATE NOT NULL,
building_id INT NOT NULL,
FOREIGN KEY (building_id) REFERENCES building(building_id) );
INSERT INTO activity(activity_name, activity_date, building_id)
VALUES ('Садко', '2022-04-16', 1), ('Король Ричард', '2021-03-15', 2), ('Щелкунчик', '2022-08-01', 3),
('Князь Игорь', '2023-01-10', 4), ('Король Ричард', '2021-03-15', 5), ('Щелкунчик', '2022-08-01', 6),
('Спящая Красавица', '2022-09-15', 7), ('Спартак', '2021-05-10', 8), ('Двенадцатая Ночь', '2023-01-04', 10),
('На Дне', '2022-10-10', 9), ('Садко', '2022-05-25', 11), ('Щелкунчик', '2022-08-25', 12),
('Музыка Души', '2023-02-22', 13), ('Трубадур', '2021-12-25', 14), ('Гамлет', '2023-10-01', 2),
```

```
('Ромео и Джульетта', '2022-02-01', 9), ('Старший Сын', '2022-11-12', 8), ('Женитьба', '2023-01-05', 8);
```

8. Создание таблицы **contest** и заполнение значений:

```
CREATE table contest (  
contest_id INTEGER PRIMARY KEY,  
contest_name VARCHAR(30),  
contest_date DATE NOT NULL );  
  
INSERT INTO contest(contest_name, contest_date)  
  
VALUES ('Пальмовая Ветвь', '2021-05-01'), ('Золотая Маска', '2021-10-01'), ('Орден Огурца',  
'2022-05-01'), ('Золотая Маска', '2022-10-01'), ('Оскар', '2021-01-02'), ('Пальмовая Ветвь',  
'2022-05-01'), ('Оскар', '2022-01-02'), ('Орден Огурца', '2021-05-01');
```

9. Создание таблицы **participant** и заполнение значений:

```
CREATE table participant (  
place INT,  
contest_id INT NOT NULL,  
artist_id INT NOT NULL,  
FOREIGN KEY(contest_id) REFERENCES contest(contest_id),  
FOREIGN KEY(artist_id) REFERENCES artist(artist_id) );  
  
INSERT INTO participant(place, contest_id, artist_id)  
  
VALUES (1, 1, 11), (2, 1, 13), (3, 1, 4), (4, 1, 1), (5, 1, 6),  
(1, 2, 17), (2, 2, 13), (3, 2, 4), (4, 2, 1), (5, 2, 6),  
(1, 3, 1), (2, 3, 2), (3, 3, 3), (4, 3, 4), (5, 3, 5),  
(1, 5, 8), (2, 5, 7), (3, 5, 6), (4, 5, 5), (5, 5, 19),  
(1, 6, 3), (2, 6, 5), (3, 6, 18), (4, 6, 11), (5, 6, 10),  
(1, 7, 20), (2, 7, 16), (3, 7, 15), (4, 7, 13), (5, 7, 1),  
(1, 8, 7), (2, 8, 2), (3, 8, 5), (4, 8, 14), (5, 8, 17);
```

5. Описание используемых запросов на языке SQL.

Система должна эффективно выдавать ответы на следующие запросы:

1. Получить перечень культурных сооружений указанного типа в целом или удовлетворяющих заданным характеристикам (например, залы, вмещающие не менее заданного числа зрителей).

```
SELECT building_name  
  
FROM building INNER JOIN type USING(type_id)  
  
WHERE capacity >= 100 AND type_name = «Концертный зал»;
```

2. Получить список артистов, выступающих в некотором жанре.

```
SELECT artist_name
FROM artist INNER JOIN genre USING(genre_id)
WHERE genre_name = 'Комедия'
ORDER BY artist_name;
```

3. Получить список артистов, работающих с некоторым импресарио.

```
SELECT artist_name
FROM artist
WHERE artist_id IN (SELECT art_imp.artist_id
FROM art_imp
WHERE art_imp.impresario_id = (SELECT impresario_id
FROM impresario
WHERE impresario_name = 'Нигай А.Ф.'))
ORDER BY artist_name
```

4. Получить список артистов, работающих более чем с одним импресарио с их указанием.

```
SELECT artist_name
FROM artist
WHERE artist_id IN (
    SELECT artist_id FROM art_imp
    GROUP BY artist_id
    HAVING count(*) > 1)
```

5. Получить список импресарио указанного артиста.

```
SELECT impresario_name
FROM impresario
WHERE impresario_id IN (
    SELECT impresario_id FROM art_imp
    WHERE artist_id = (
        SELECT artist_id FROM artist WHERE artist_name = 'Арбузов В.О.') );
```

6. Получить перечень концертных мероприятий, проведенных в течение заданного периода времени в целом либо указанным организатором.

```
SELECT activity_name, activity_date
FROM activity
WHERE activity_date BETWEEN '2022-01-01' AND '2023-01-01'
ORDER BY 2;
```

7. Получить список призеров указанного конкурса.

```
SELECT artist_name AS 'Имя', place AS 'Место', contest_name AS 'Награда',  
contest_date AS 'Дата'  
  
FROM artist INNER JOIN participant USING(artist_id)  
  
INNER JOIN contest USING(contest_id)  
  
WHERE contest_name = 'Орден Огурца' AND place < 4  
  
ORDER BY 3;
```

8. Получить перечень концертных мероприятий, проведенных в указанном культурном сооружении.

```
SELECT activity_name, building_name  
FROM activity INNER JOIN building USING(building_id)  
WHERE building_name = 'Театр Маяковского'
```

9. Получить список импресарио определенного жанра.

```
SELECT impresario_name, genre_name  
FROM impresario INNER JOIN genre USING(genre_id)  
WHERE genre_name = 'Комедия'
```

10. Получить список артистов, участвовавших в конкурсах в течение определенного периода времени.

```
SELECT artist_name AS 'Имя'  
FROM artist  
WHERE artist_id IN (  
    SELECT artist_id FROM participant WHERE contest_id IN (  
        SELECT contest_id FROM contest WHERE contest_date > '2022-03-01') );
```

11. Получить список организаторов культурных мероприятий и число проведенных ими концертов в течение определенного периода времени.

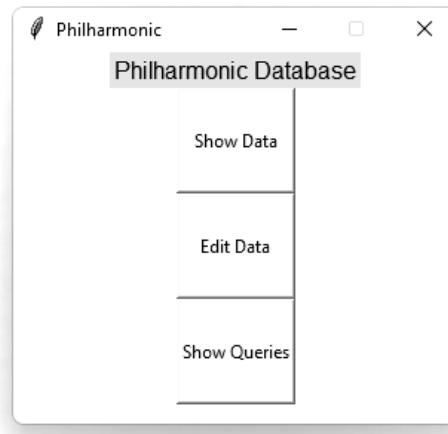
```
SELECT building_name, count(activity_name) AS 'Число мероприятий', activity_date  
AS 'Дата'  
  
FROM activity INNER JOIN building USING(building_id)  
  
GROUP BY building_name  
  
HAVING activity_date > '2022-12-31'
```

12. Получить перечень культурных сооружений, а также даты проведения на них культурных мероприятий в течение определенного периода времени.

```
SELECT building_name, activity_date AS 'Дата', activity_name AS 'Мероприятие'  
FROM activity INNER JOIN building USING(building_id)  
WHERE activity_date BETWEEN '2022-12-31' AND '2023-12-31'  
  
ORDER BY 2;
```

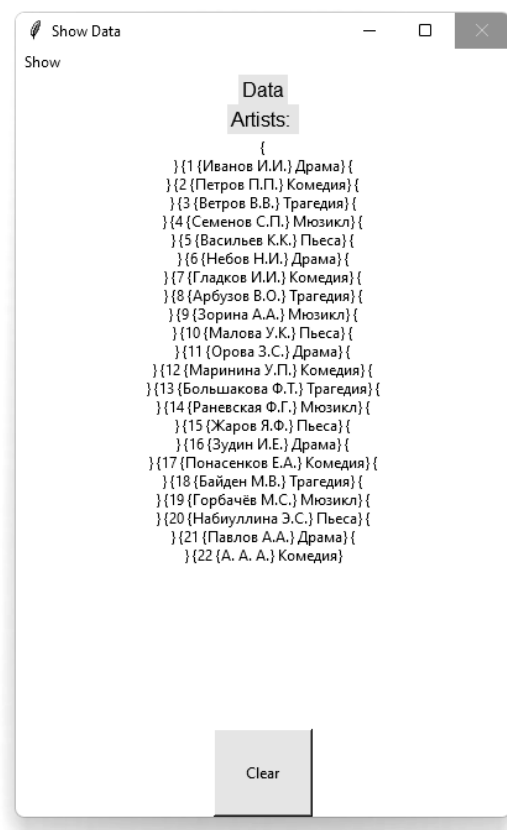

6. Руководство пользователя по работе с приложением.

1. На входном экране выбрать один из вариантов пользования системой.



2. При выборе Show Data можно просмотреть данные из БД.

2.1. По кнопке Show необходимо выбрать категорию данных, которые надо просмотреть. Затем нажать на Clear и очистить окно для просмотра следующих данных.



3. В пункте **Edit Data** можно создать (**Create**) или удалить (**Delete**) элемент из таблиц данных.

3.1. При выборе **Create – Artists** в поля ввода информации необходимо ввести данные о новом артисте и нажать **Add** для добавления его в БД.

The screenshot shows a window titled "Edit Data" with a menu bar containing "Create" and "Delete". A dropdown menu is open under "Create", showing three options: "Artists" (highlighted), "Impresario", and "Activity". An "Edit" button is visible in the top right corner of the main area.

The screenshot shows the "Edit Data" window with the "Create" menu closed. The form contains the following fields: "Id" with the value "23", "Name" with the value "Новичков Н.Н.", and "Genre Name" with the value "Комедия". An "Add" button is located at the bottom right of the window.

The screenshot shows the "Edit Data" window with the "Create" menu closed. The form contains the "Name" field with the value "Новичков Н.Н.". A "Delete" button is located at the bottom right of the window.

3.2. При выборе пункта **Delete – Artists** в поле ввода надо ввести имя артиста и нажать **Delete** для его удаления из базы данных.

4. При выборе пункта **Show Queries** необходимо во вкладке **Queries** выбрать номер запроса.

Show Queries

Queries

1
2
3
4
5
6
7
8
9
10
11
12

Запрос

Clear

4.1 При выборе запроса необходимо ввести данные для запроса вручную. Затем нажать на «**Отправить**» и получить ответ. После нажать **Clear** для очищения поля.

Show Queries

Queries

Запрос

12. Получить перечень культурных сооружений,
а также даты проведения на них культурных мероприятий в течение определенного периода времени.

От периода [ГГГГ-ММ-ДД]:
2022-05-10

до периода [ГГГГ-ММ-ДД]:
2022-12-03

{
} {{Культурный Центр} 2022-05-25 Садко} {
}
} {{Студия им. Моргенштерна} 2022-07-01 РокОркестра} {
}
} {{Консерватория им. Чайковского} 2022-08-01 Щелкунчик} {
}
} {{Консерватория им. Гнесина} 2022-08-01 Щелкунчик} {
}
} {{Дом Творчества} 2022-08-25 Щелкунчик} {
}
} {{Театр Мюзикла} 2022-09-15 {Спящая Красавица}} {
}
} {{Театр Моссовета} 2022-10-10 {На Дне}} {
}
} {{Театр Маяковского} 2022-11-12 {Старший Сын}} {
}
} {{Консерватория им. Гнесина} 2022-12-05 {Лебединое Озеро}} {
}

Отправить

Clear