

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Базы данных»
Тема: Реализация базы данных в СУБД PostgreSQL

Студент гр. 1303

Самохин К.А.

Преподаватель

Заславский М.М.

Санкт-Петербург

2023

Цель работы.

Развернуть локально PostgreSQL, написать запросы для создания и заполнения таблиц, написать запросы к БД, отвечающие на вопросы в задания.

Задание.

Вариант 18

Пусть требуется создать программную систему, предназначенную для менеджера музыкальных групп. Такая система должна обеспечивать хранение сведений о группах, включающих название группы, год образования и страну, состав исполнителей, положение в последнем хит-параде; репертуар группы. Сведения о каждой песне из репертуара группы — это ее название, композитор, автор текста. Необходимо также хранить данные о последней гастрольной поездке каждой группы: название гастрольной программы, названия населенных пунктов, дата начала и окончания выступлений, средняя цена билета (зависит от места выступления и положения группы в хит-параде). Возможно появление новой группы и изменение состава исполнителей. Каждая песня может быть в репертуаре только одной группы. Менеджеру могут потребоваться следующие сведения:

- Автор текста, композитор и дата создания песни с данным названием? В репертуар какой группы она входит?
- Репертуар наиболее популярной группы?
- Цена билета на последний концерт указанной группы?
- Состав исполнителей группы с заданным названием, их возраст и амплуа?
- Место и продолжительность гастролей группы с заданным названием?
- Какие группы в текущем году отмечают юбилей
- Самый молодой вокалист? Какую группу он представляет?

Выполнение работы.

Создана база данных.

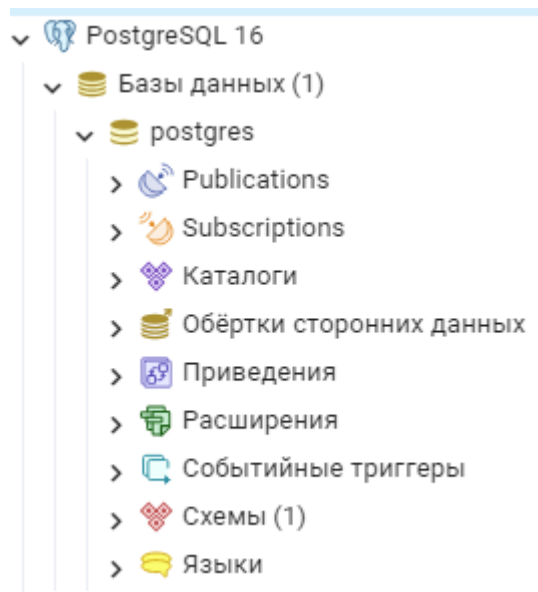


Рисунок 1. Вид созданной БД.

Созданы таблицы «Группа», «Артист», «Гастроли», «Артист_группа», «Страна», «Человек», «Город», «Песня» в соответствии со структурой БД, составленной в первой лабораторной работе, и написаны запросы на их заполнение.

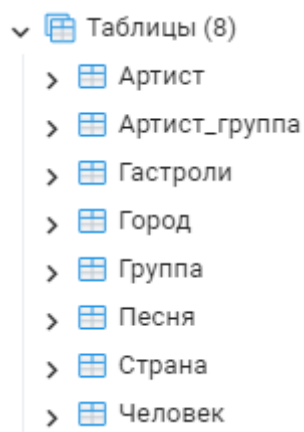


Рисунок 2. Созданные таблицы.

Работа запросов, необходимых менеджеру:

Запрос История запросов

```
1 SELECT Название_песни, Композитор.ФИО AS Композитор, Автор.ФИО AS Автор_текста, Дат
2 JOIN Группа ON Группа.id_группы = Песня.id_группы
3 JOIN Человек AS Композитор ON Песня.Композитор = Композитор.id_человека
4 JOIN Человек AS Автор ON Песня.Автор_песни = Автор.id_человека
5 WHERE Название_песни = 'Auslander';
6
```

Data Output Сообщения Notifications

	Название_песни text	Композитор text	Автор_текста text	Дата_написания date	Название_гру text
1	Auslander	Michael Walter	Hans Schwarz	2000-09-12	Wolpertinger

Рисунок 3. Автор текста, композитор и дата создания песни с данным названием, группа, в репертуар которой она входит.

Запрос История запросов

```
1 SELECT Название_песни FROM Песня
2 JOIN Группа ON Группа.id_группы = Песня.id_группы
3 WHERE Положение_в_хит_параде = 1;
4
```

Data Output Сообщения Notifications

	Название_песни text
1	Auslander
2	Sonne

Рисунок 4. Репертуар наиболее популярной группы.

Запрос История запросов

```

1 SELECT Название_группы, Название_программы, Название_города, Дата_окончания - Дата_
2     JOIN Город ON Город.id_города = Гастроли.id_города
3     JOIN Группа ON Группа.id_группы = Гастроли.id_группы
4 WHERE Название_группы = 'Арт-обстрел';

```

Data Output Сообщения Notifications

	Название_группы text	Название_программы text	Название_города text	Продолжительность integer
1	Арт-обстрел	Я русский!	Новосибирск	12
2	Арт-обстрел	Я русский!	Краснодар	12

Рисунок 7. Место и продолжительность гастролей группы с заданным названием.

Запрос История запросов

```

1 SELECT Название_группы, Год_создания FROM Группа
2 WHERE (DATE_PART('year', CURRENT_DATE) - Год_создания) IN (5, 10, 15, 20, 25, 30);

```

Data Output Сообщения Notifications

	Название_группы text	Год_создания integer
1	Wolpertinger	1998

Рисунок 8. Какие группы в текущем году отмечают юбилей.

Запрос

История запросов

1 SELECT Название_группы, ФИО, Возраст FROM Артист_группа
2 JOIN Группа ON Группа.id_группы = Артист_группа.id_группы
3 JOIN Артист ON Артист.id_артиста = Артист_группа.id_артиста
4 JOIN Человек ON Артист.id_человека = Человек.id_человека
5 WHERE Возраст IN (SELECT MIN(Возраст) FROM Артист WHERE Амплуа = 'Вокалист');

Data Output

Сообщения

Notifications

	Название_группы text	ФИО text	Возраст integer
1	Deity of Fallen	Баранов Сергей Михайлович	22

Рисунок 9. Самый молодой вокалист и группа, которую он представляет.

Вывод.

В ходе выполнения работы был развернут локально PostgreSQL, написаны запросы для создания и заполнения таблиц в соответствии со структурой БД, написаны запросы к БД, отвечающие на вопросы в заданиях.

Приложение А

Ссылки

Pull Request: <https://github.com/moevm/sql-2023-1303/pull/37>

DB Fiddle: <https://www.db-fiddle.com/f/pgVbQ3hwzpwBWYLFU6tbzA/11>

Приложение Б

Исходный код

Файл lab2.sql:

```
CREATE TABLE Страна (
    id_страны SMALLSERIAL PRIMARY KEY,
    Название_страны TEXT NOT NULL
);

CREATE TABLE Группа (
    id_группы SMALLSERIAL PRIMARY KEY,
    Название_группы TEXT NOT NULL,
    Год_создания SMALLINT NOT NULL,
    id_страны SMALLINT,
    Положение_в_хит_параде SMALLSERIAL NOT NULL,
    FOREIGN KEY (id_страны) REFERENCES Страна(id_страны) ON
DELETE SET NULL
);

CREATE TABLE Человек (
    id_человека SMALLSERIAL PRIMARY KEY,
    ФИО TEXT NOT NULL
);

CREATE TABLE Песня (
    id_песни SMALLSERIAL PRIMARY KEY,
    Название_песни TEXT NOT NULL,
    Композитор SMALLINT,
    FOREIGN KEY (Композитор) REFERENCES Человек(id_человека) ON
DELETE SET NULL,
    Автор_песни SMALLINT,
    FOREIGN KEY (Автор_песни) REFERENCES Человек(id_человека) ON
DELETE SET NULL,
    id_группы SMALLINT,
```

```

        FOREIGN KEY (id_группы) REFERENCES Группа(id_группы) ON
DELETE CASCADE,
        Дата_написания DATE NOT NULL
);

CREATE TABLE Артист(
        id_артиста SMALLSERIAL PRIMARY KEY,
        Возраст INTEGER NOT NULL,
        Амплуа TEXT NOT NULL,
        id_человека SMALLINT,
        FOREIGN KEY (id_человека) REFERENCES Человек(id_человека) ON
DELETE CASCADE
);

CREATE TABLE Город(
        id_города SMALLSERIAL PRIMARY KEY,
        Название_города TEXT NOT NULL
);

CREATE TABLE Артист_группа(
        id_группы SMALLINT,
        id_артиста SMALLINT,
        PRIMARY KEY (id_группы, id_артиста)
);

CREATE TABLE Гастроли(
        id_выступления SMALLSERIAL PRIMARY KEY,
        Название_программы TEXT NOT NULL,
        id_города SMALLINT,
        Дата_начала DATE NOT NULL,
        Дата_окончания DATE NOT NULL,
        Дата_выступления DATE NOT NULL,
        Цена_билета NUMERIC NOT NULL,
        id_группы SMALLINT,

```

```

        FOREIGN KEY (id_города) REFERENCES Город(id_города) ON DELETE
CASCADE,
        FOREIGN KEY (id_группы) REFERENCES Группа(id_группы) ON
DELETE CASCADE
);

```

```

INSERT INTO Страна(Название_страны) VALUES
    ('Россия'),
    ('Германия'),
    ('Ведикобритания');

```

```

INSERT INTO Человек(ФИО) VALUES
    ('Баранов Сергей Михайлович'),
    ('Иванов Георгий Алексеевич'),
    ('Michael Walter'),
    ('Hans Schwarz'),
    ('Peter Schmidt'),
    ('John Smith'),
    ('Thomas Davies'),
    ('Henry Walker');

```

```

INSERT INTO Город(Название_города) VALUES
    ('Москва'),
    ('Санкт-Петербург'),
    ('Краснодар'),
    ('Новосибирск'),
    ('Мюнхен'),
    ('Берлин'),
    ('Лондон'),
    ('Манчестер');

```

```

INSERT INTO Группа(Название_группы, Год_создания, id_страны,
Положение_в_хит_параде) VALUES
    ('Dead end', 1997, 3, 2),
    ('Арт-обстрел', 2000, 1, 3),

```

```

('Wolpertinger', 1995, 2, 1),
('Мираж', 2004, 1, 4);

INSERT INTO Песня(Название_песни, Композитор, Автор_песни,
id_группы, Дата_написания) VALUES
('Rainy Day', 2, 1, 4, '2009-03-25'),
('Time Bomb', 2, 1, 4, '2010-01-17'),
('Auslander', 3, 4, 3, '2000-09-12'),
('Sonne', 3, 4, 3, '2003-11-04'),
('Amnesia', 6, 5, 1, '2001-06-15'),
('After Dark', 6, 5, 1, '2008-10-20'),
('Мой друг', 8, 7, 2, '2010-08-18'),
('Тепло', 8, 7, 2, '2019-04-24');

INSERT INTO Артист(Возраст, Амплуа, id_человека) VALUES
(22, 'Вокалист', 1),
(20, 'Гитарист', 2),
(24, 'Волкалист', 8),
(21, 'Басист', 7),
(25, 'Вокалист', 3),
(24, 'Барабанщик', 4),
(23, 'Гитарист', 5),
(27, 'Барабанщик', 6);

INSERT INTO Артист_группа (id_группы, id_артиста) VALUES
(4, 1),
(4, 2),
(1, 5),
(1, 6),
(2, 7),
(2, 8),
(3, 3),
(3, 4);

```

```

INSERT INTO Гастроли(Название_программы, id_города, Дата_начала,
Дата_окончания, Дата_выступления, Цена_билета, id_группы) VALUES
    ('Tour', 7, '2023-04-12', '2023-04-22', '2023-04-15', 999.99,
1),
    ('Tour', 8, '2023-04-12', '2023-04-22', '2023-04-19', 899.99,
1),
    ('Я русский!', 4, '2023-06-23', '2023-07-05', '2023-06-27',
499.99, 2),
    ('Я русский!', 3, '2023-06-23', '2023-07-05', '2023-07-01',
399.99, 2),
    ('Flugzeug', 5, '2023-02-03', '2023-02-10', '2023-02-04',
1199.99, 3),
    ('Flugzeug', 6, '2023-02-03', '2023-02-10', '2023-02-08',
1399.99, 3),
    ('Малый тур', 2, '2023-09-10', '2023-09-21', '2023-09-13',
799.99, 4),
    ('Малый тур', 1, '2023-09-10', '2023-09-21', '2023-02-19',
799.99, 4);

```

```

SELECT Название_песни, Композитор.ФИО AS Композитор, Автор.ФИО AS
Автор_текста, Дата_написания, Название_группы FROM Песня
    JOIN Группа ON Группа.id_группы = Песня.id_группы
    JOIN Человек AS Композитор ON Песня.Композитор =
Композитор.id_человека
    JOIN Человек AS Автор ON Песня.Автор_песни =
Автор.id_человека
WHERE Название_песни = 'Auslander';

```

```

SELECT Название_песни FROM Песня
    JOIN Группа ON Группа.id_группы = Песня.id_группы
WHERE Положение_в_хит_параде = 1;

```

```

SELECT Название_группы, Название_программы, Дата_выступления,
Цена_билета FROM Гастроли
    JOIN Группа ON Группа.id_группы = Гастроли.id_группы

```

```
WHERE Название_группы = 'Wolpertinger' AND Дата_выступления IN  
(SELECT MAX(Дата_выступления) FROM Гастроли GROUP BY id_группы);
```

```
SELECT Название_группы, ФИО, Возраст, Амплуа FROM Артист_группа  
    JOIN Группа ON Группа.id_группы = Артист_группа.id_группы  
    JOIN Артист ON Артист.id_артиста = Артист_группа.id_артиста  
    JOIN Человек ON Артист.id_человека = Человек.id_человека  
WHERE Название_группы = 'Dead end'  
ORDER BY ФИО ASC;
```

```
SELECT Название_группы, Название_программы, Название_города,  
Дата_окончания - Дата_начала AS Продолжительность FROM Гастроли  
    JOIN Город ON Город.id_города = Гастроли.id_города  
    JOIN Группа ON Группа.id_группы = Гастроли.id_группы  
WHERE Название_группы = 'Арт-обстрел';
```

```
SELECT Название_группы, Год_создания FROM Группа  
WHERE CAST((DATE_PART('year', CURRENT_DATE) - Год_создания) AS  
INTEGER)%5 = 0;
```

```
SELECT Название_группы, ФИО, Возраст FROM Артист_группа  
    JOIN Группа ON Группа.id_группы = Артист_группа.id_группы  
    JOIN Артист ON Артист.id_артиста = Артист_группа.id_артиста  
    JOIN Человек ON Артист.id_человека = Человек.id_человека  
WHERE Возраст IN (SELECT MIN(Возраст) FROM Артист WHERE Амплуа =  
'Вокалист');
```