**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**

| Студентка гр. 2384 |  | Соц Е.А. |
| --- | --- | --- |
| Преподаватель |  | Заславский М.М. |

Санкт-Петербург

2024

## **Цель работы**

Реализовать ранее спроектированную базу данных в СУДБ PostgreSQL. Создать и заполнить тестовыми данными таблицы. Создать запросы, отвечающие на поставленные вопросы.

## **Задание**

Вариант 18

* Написать запросы для создания таблиц из предыдущей лабораторный работы
* Заполнить тестовыми данными: 5-10 строк на каждую таблицу, обязательно наличие связи между ними, данные приближены к реальности.
* Написать запросы к БД, отвечающие на вопросы:

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

* Исходный код выложить на [www.db-fiddle.com](http://www.db-fiddle.com) для проверки работоспособности
* Исходный код в виде .sql файла запушить в виде PR в репо

## **Выполнение работы**

В ходе выполнения первой лабораторной работы была описана структура базы данных, предназначенная для менеджера музыкальных групп:

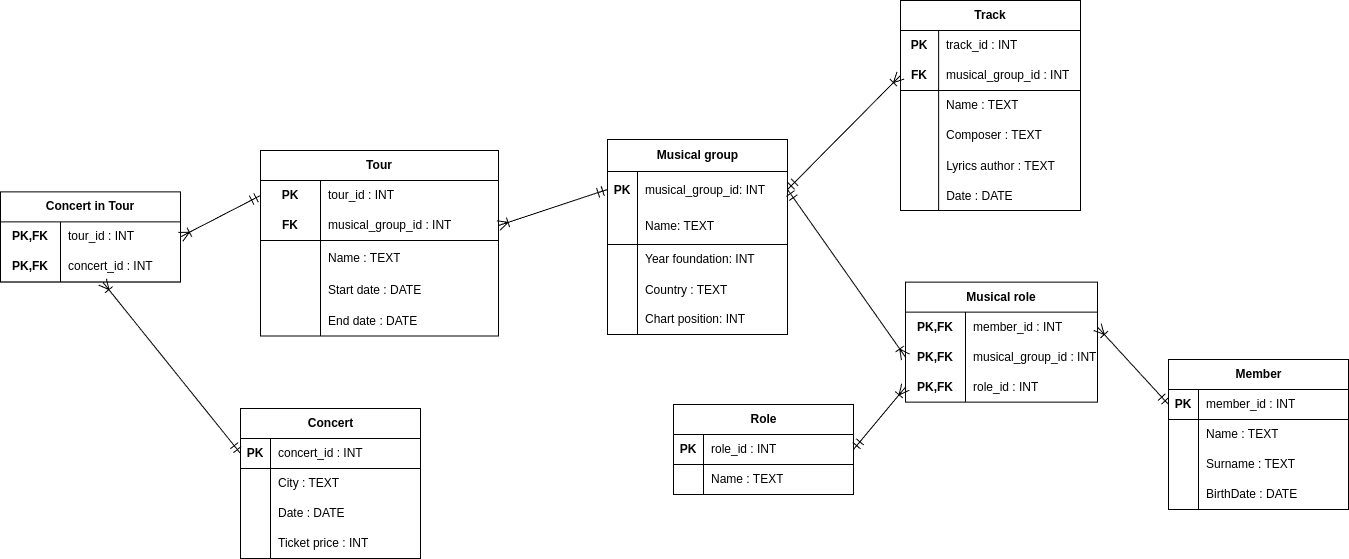


Рисунок 1 – структура БД

Для выполнения данной лабораторной работы был установлен PostgreSQL. Были написаны SQL-запросы для создания таблиц, соответствующих спроектированной базе данных. В таблицах реализованы связи с помощью соответствующих ключей. Была произведена вставка тестовыми данными в таблицы.

Наполнение таблиц можно посмотреть с помощью запроса SELECT \* FROM <name\_table>. Ниже будут приведены снимки экрана, отражающие наполненность каждой таблицы.

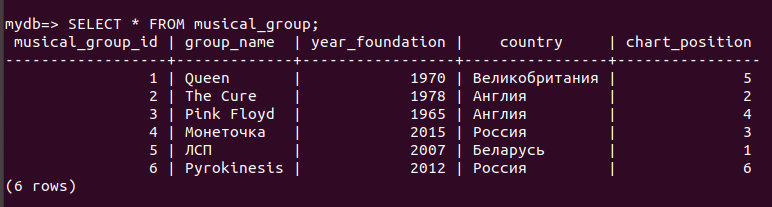


Рисунок 2 – Таблица musical\_group

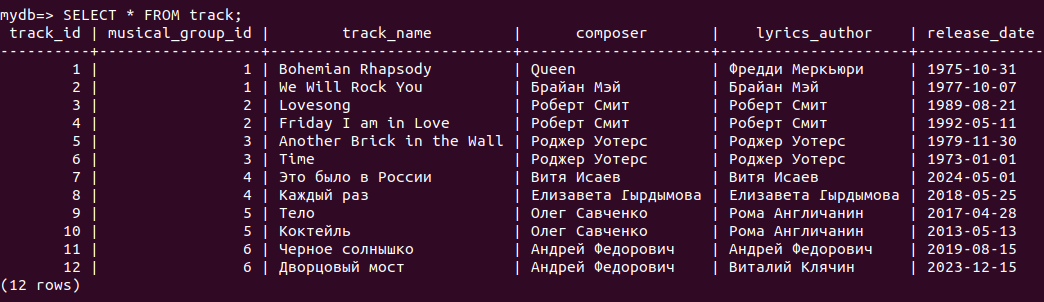


Рисунок 3 – Таблица track



Рисунок 4 – Таблица member

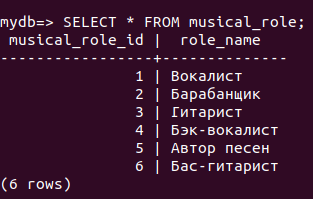


Рисунок 5 – Таблица musical\_role

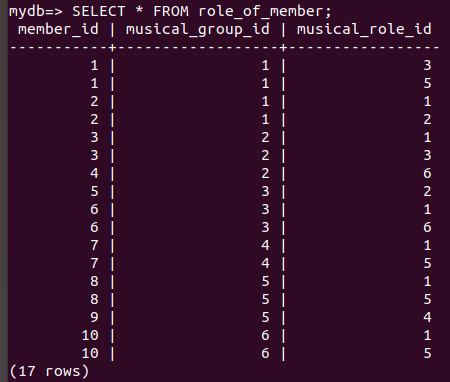


Рисунок 6 – Таблица role\_of\_member

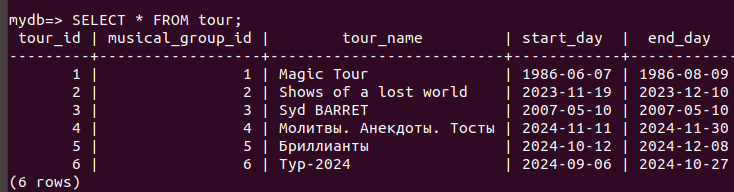


Рисунок 7 – Таблица tour

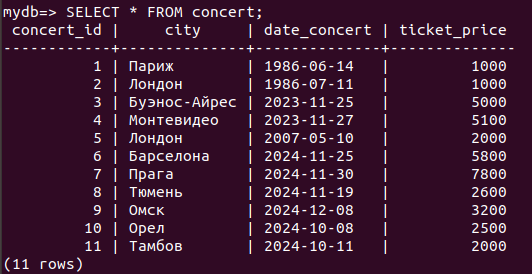


Рисунок 8 – Таблица concert

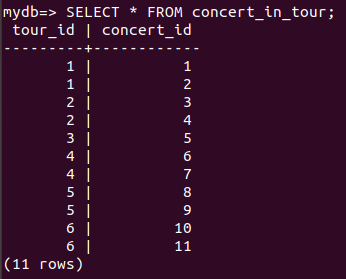


Рисунок 9 – Таблица concert\_in\_tour

Теперь выполним SQL-запросы из задания:

1. Автор текста, композитор и дата создания песни с данным названием? В репертуар какой группы она входит?

Для теста была выбрана песня “Дворцовый мост”.

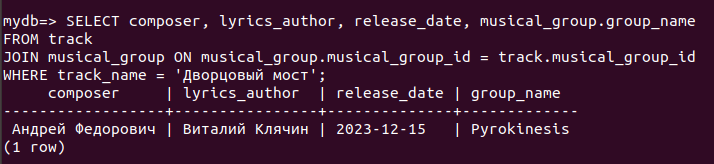


Рисунок 10 – Информация об определенной песне

1. Репертуар наиболее популярной группы?

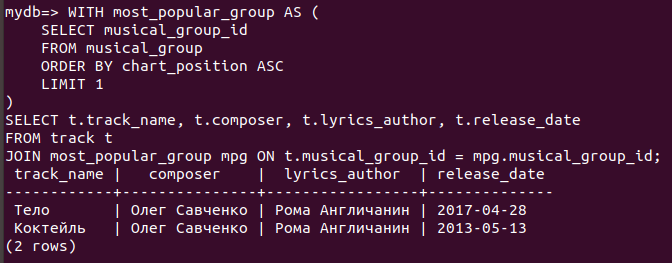


Рисунок 11 – Репертуар самой популярной группы

1. Цена билета на последний концерт указанной группы?

Для теста была выбрана группа “Монеточка”.

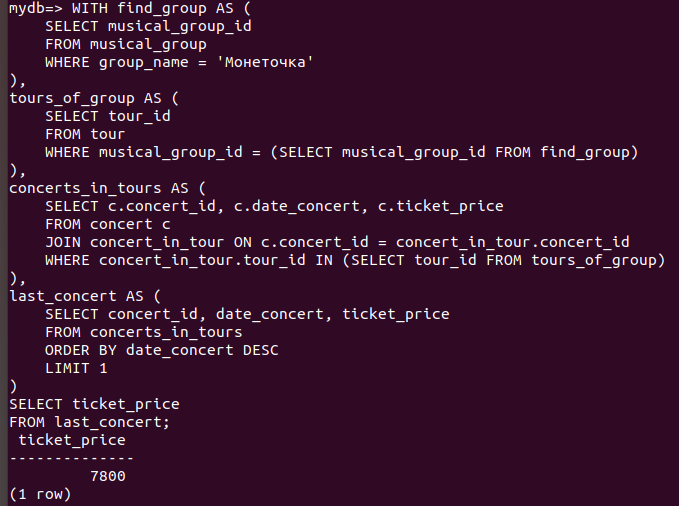


Рисунок 12 – Цена билета

1. Состав исполнителей группы с заданным названием, их возраст и амплуа?

Для теста была выбрана группа “The Cure”.

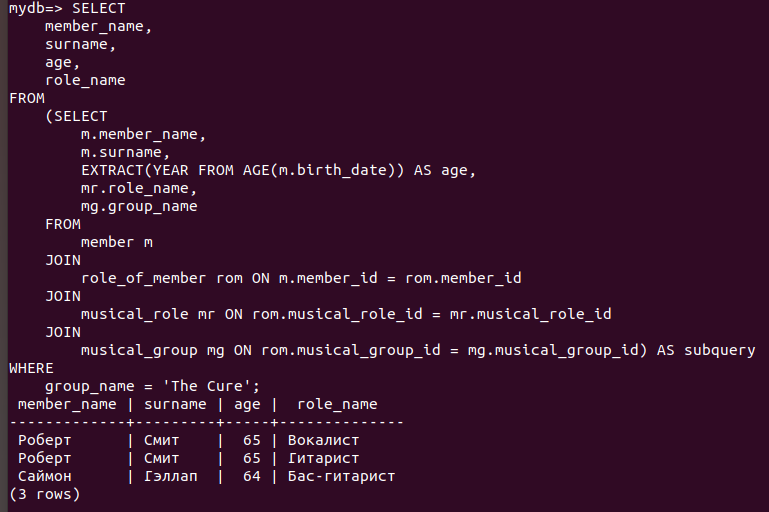


Рисунок 13 – Информация о составе определенной группы

1. Место и продолжительность гастролей группы с заданным названием?

Для теста была выбрана группа “ЛСП”.

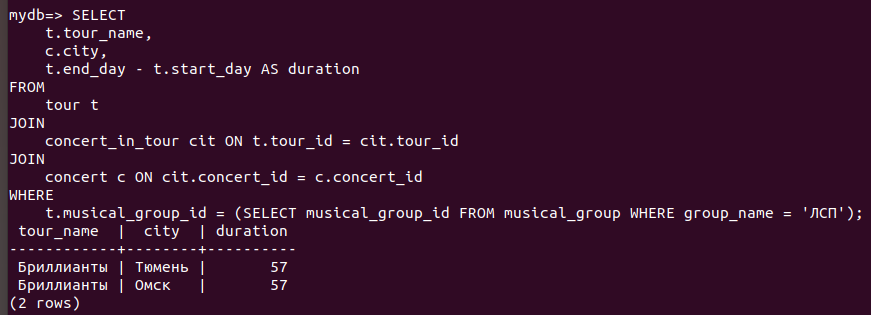


Рисунок 14 – Информация о турах заданной группы

1. Какие группы в текущем году отмечают юбилей?

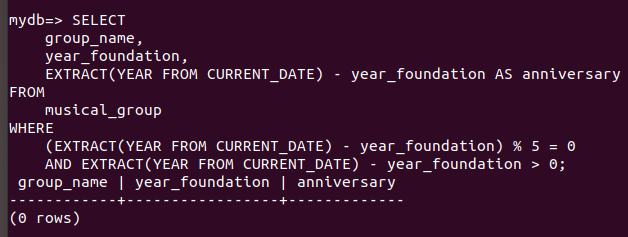


Рисунок 15 – Юбилей в текущем году

Никакой информации не получено, но это правильный ответ, ведь по моим тестовым данным никакая группа не отмечает юбилей текущем году.

Для корректной проверки запроса можно задать какой-то конкретный год юбилея:

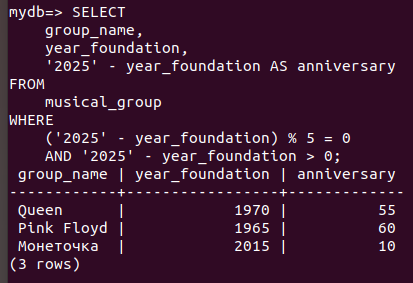


Рисунок 16 – Юбилей в конкретном году

1. Самый молодой вокалист? Какую группу он представляет?

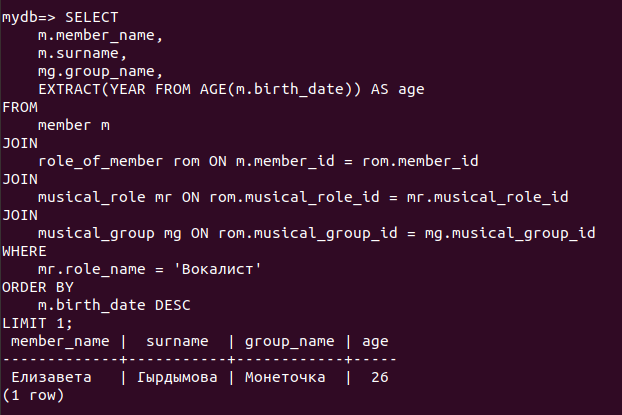


Рисунок 17 – Самый молодой вокалист

В приложении А представлен исходный код.

В приложении В находится ссылка на исходный код в DB-Fiddle.

В приложении С представлена ссылка на PR.

## **Вывод**

В ходе выполнения лабораторной работы была реализована ранее спроектированная БД. Были созданы таблицы, наполнены тестовыми данными и реализованы SQL-запросы в соответствии с вариантом.

## **ПРИЛОЖЕНИЕ А**

**ИСХОДНЫЙ КОД**

DROP TABLE IF EXISTS musical\_group, track, role\_of\_member, member, musical\_role, tour, concert\_in\_tour, concert;

CREATE TABLE musical\_group (

musical\_group\_id SERIAL PRIMARY KEY,

group\_name VARCHAR(50) NOT NULL,

year\_foundation INT NOT NULL,

country VARCHAR(30) NOT NULL,

chart\_position INT NOT NULL

);

CREATE TABLE track (

track\_id SERIAL PRIMARY KEY,

musical\_group\_id INT NOT NULL,

track\_name VARCHAR(50) NOT NULL,

composer VARCHAR(50) NOT NULL,

lyrics\_author VARCHAR(50) NOT NULL,

release\_date DATE NOT NULL,

FOREIGN KEY (musical\_group\_id) REFERENCES musical\_group(musical\_group\_id)

);

CREATE TABLE member (

member\_id SERIAL PRIMARY KEY,

member\_name VARCHAR(20) NOT NULL,

surname VARCHAR(20) NOT NULL,

birth\_date DATE NOT NULL

);

CREATE TABLE musical\_role (

musical\_role\_id SERIAL PRIMARY KEY,

role\_name VARCHAR(20) NOT NULL

);

CREATE TABLE role\_of\_member (

member\_id INT NOT NULL,

musical\_group\_id INT NOT NULL,

musical\_role\_id INT NOT NULL,

PRIMARY KEY (member\_id, musical\_group\_id, musical\_role\_id),

FOREIGN KEY (member\_id) REFERENCES member(member\_id),

FOREIGN KEY (musical\_group\_id) REFERENCES musical\_group(musical\_group\_id),

FOREIGN KEY (musical\_role\_id) REFERENCES musical\_role(musical\_role\_id)

);

CREATE TABLE tour (

tour\_id SERIAL PRIMARY KEY,

musical\_group\_id INT NOT NULL,

tour\_name VARCHAR(30) NOT NULL,

start\_day DATE NOT NULL,

end\_day DATE NOT NULL,

FOREIGN KEY (musical\_group\_id) REFERENCES musical\_group(musical\_group\_id)

);

CREATE TABLE concert (

concert\_id SERIAL PRIMARY KEY,

city VARCHAR(30) NOT NULL,

date\_concert DATE NOT NULL,

ticket\_price INT NOT NULL

);

CREATE TABLE concert\_in\_tour (

tour\_id INT NOT NULL,

concert\_id INT NOT NULL,

PRIMARY KEY (tour\_id, concert\_id),

FOREIGN KEY (tour\_id) REFERENCES tour(tour\_id),

FOREIGN KEY (concert\_id) REFERENCES concert(concert\_id)

);

INSERT INTO musical\_group(group\_name, year\_foundation, country, chart\_position)

VALUES

('Queen', '1970', 'Великобритания', 5),

('The Cure', '1978', 'Англия', 2),

('Pink Floyd', '1965', 'Англия', 4),

('Монеточка', '2015', 'Россия', 3),

('ЛСП', '2007', 'Беларусь', 1),

('Pyrokinesis', '2012', 'Россия', 6);

INSERT INTO track(musical\_group\_id, track\_name, composer, lyrics\_author,release\_date)

VALUES

(1, 'Bohemian Rhapsody', 'Queen', 'Фредди Меркьюри', '1975-10-31'),

(1, 'We Will Rock You', 'Брайан Мэй', 'Брайан Мэй', '1977-10-07'),

(2, 'Lovesong', 'Роберт Смит', 'Роберт Смит', '1989-08-21'),

(2, 'Friday I am in Love', 'Роберт Смит', 'Роберт Смит', '1992-05-11'),

(3, 'Another Brick in the Wall', 'Роджер Уотерс', 'Роджер Уотерс', '1979-11-30'),

(3, 'Time', 'Роджер Уотерс', 'Роджер Уотерс', '1973-01-01'),

(4, 'Это было в России', 'Витя Исаев', 'Витя Исаев', '2024-05-01'),

(4, 'Каждый раз', 'Елизавета Гырдымова', 'Елизавета Гырдымова', '2018-05-25'),

(5, 'Тело', 'Олег Савченко', 'Рома Англичанин', '2017-04-28'),

(5, 'Коктейль', 'Олег Савченко', 'Рома Англичанин', '2013-05-13'),

(6, 'Черное солнышко', 'Андрей Федорович', 'Андрей Федорович', '2019-08-15'),

(6, 'Дворцовый мост', 'Андрей Федорович', 'Виталий Клячин', '2023-12-15');

INSERT INTO member(member\_name, surname, birth\_date)

VALUES

('Брайн', 'Мэй', '1947-07-19'),

('Роджер', 'Тейлор', '1949-07-26'),

('Роберт', 'Смит', '1959-04-21'),

('Саймон', 'Гэллап', '1960-06-01'),

('Ник', 'Мейсон', '1944-01-27'),

('Роджер', 'Уотерс', '1943-09-06'),

('Елизавета', 'Гырдымова', '1998-06-01'),

('Олег', 'Савченко', '1989-07-10'),

('Петр', 'Клюев', '1989-05-28'),

('Андрей', 'Федорович', '1995-12-16');

INSERT INTO musical\_role(role\_name)

VALUES

('Вокалист'),

('Барабанщик'),

('Гитарист'),

('Бэк-вокалист'),

('Автор песен'),

('Бас-гитарист');

INSERT INTO role\_of\_member(member\_id, musical\_group\_id, musical\_role\_id)

VALUES

(1, 1, 3),

(1, 1, 5),

(2, 1, 1),

(2, 1, 2),

(3, 2, 1),

(3, 2, 3),

(4, 2, 6),

(5, 3, 2),

(6, 3, 1),

(6, 3, 6),

(7, 4, 1),

(7, 4, 5),

(8, 5, 1),

(8, 5, 5),

(9, 5, 4),

(10, 6, 1),

(10, 6, 5);

INSERT INTO tour(musical\_group\_id, tour\_name, start\_day, end\_day)

VALUES

(1, 'Magic Tour', '1986-06-07', '1986-08-09'),

(2, 'Shows of a lost world', '2023-11-19', '2023-12-10'),

(3, 'Syd BARRET', '2007-05-10', '2007-05-10'),

(4, 'Молитвы. Анекдоты. Тосты', '2024-11-11', '2024-11-30'),

(5, 'Бриллианты', '2024-10-12', '2024-12-08'),

(6, 'Тур-2024', '2024-09-06', '2024-10-27');

INSERT INTO concert(city, date\_concert, ticket\_price)

VALUES

('Париж', '1986-06-14', 1000),

('Лондон', '1986-07-11', 1000),

('Буэнос-Айрес', '2023-11-25', 5000),

('Монтевидео', '2023-11-27', 5100),

('Лондон', '2007-05-10', 2000),

('Барселона', '2024-11-25', 5800),

('Прага', '2024-11-30', 7800),

('Тюмень', '2024-11-19', 2600),

('Омск', '2024-12-08', 3200),

('Орел', '2024-10-08', 2500),

('Тамбов', '2024-10-11', 2000);

INSERT INTO concert\_in\_tour(tour\_id, concert\_id)

VALUES

(1, 1),

(1, 2),

(2, 3),

(2, 4),

(3, 5),

(4, 6),

(4, 7),

(5, 8),

(5, 9),

(6, 10),

(6, 11);

-- 1 Автор текста, композитор и дата создани песни с данным названием? В репертуар какой группы входит?--

SELECT composer, lyrics\_author, release\_date, musical\_group.group\_name

FROM track

JOIN musical\_group ON musical\_group.musical\_group\_id = track.musical\_group\_id

WHERE track\_name = 'Дворцовый мост';

--2 репертуар наиболее популярной группы--

WITH most\_popular\_group AS (

SELECT musical\_group\_id

FROM musical\_group

ORDER BY chart\_position ASC

LIMIT 1

)

SELECT t.track\_name, t.composer, t.lyrics\_author, t.release\_date

FROM track t

JOIN most\_popular\_group mpg ON t.musical\_group\_id = mpg.musical\_group\_id;

--3 цена билета на последний концерт указанной группы

WITH find\_group AS (

SELECT musical\_group\_id

FROM musical\_group

WHERE group\_name = 'Монеточка'

),

tours\_of\_group AS (

SELECT tour\_id

FROM tour

WHERE musical\_group\_id = (SELECT musical\_group\_id FROM find\_group)

),

concerts\_in\_tours AS (

SELECT c.concert\_id, c.date\_concert, c.ticket\_price

FROM concert c

JOIN concert\_in\_tour ON c.concert\_id = concert\_in\_tour.concert\_id

WHERE concert\_in\_tour.tour\_id IN (SELECT tour\_id FROM tours\_of\_group)

),

last\_concert AS (

SELECT concert\_id, date\_concert, ticket\_price

FROM concerts\_in\_tours

ORDER BY date\_concert DESC

LIMIT 1

)

SELECT ticket\_price

FROM last\_concert;

--4 состав группы с заданным названием, их возраст и роль

SELECT

member\_name,

surname,

age,

role\_name

FROM

(SELECT

m.member\_name,

m.surname,

EXTRACT(YEAR FROM AGE(m.birth\_date)) AS age,

mr.role\_name,

mg.group\_name

FROM

member m

JOIN

role\_of\_member rom ON m.member\_id = rom.member\_id

JOIN

musical\_role mr ON rom.musical\_role\_id = mr.musical\_role\_id

JOIN

musical\_group mg ON rom.musical\_group\_id = mg.musical\_group\_id) AS subquery

WHERE

group\_name = 'The Cure';

--5 место и продолжительность гастролей группы с заданным названием

SELECT

t.tour\_name,

c.city,

t.end\_day - t.start\_day AS duration

FROM

tour t

JOIN

concert\_in\_tour cit ON t.tour\_id = cit.tour\_id

JOIN

concert c ON cit.concert\_id = c.concert\_id

WHERE

t.musical\_group\_id = (SELECT musical\_group\_id FROM musical\_group WHERE group\_name = 'ЛСП');

--6 группы, отмечающие в текущем году юбилей

SELECT

group\_name,

year\_foundation,

EXTRACT(YEAR FROM CURRENT\_DATE) - year\_foundation AS anniversary

FROM

musical\_group

WHERE

(EXTRACT(YEAR FROM CURRENT\_DATE) - year\_foundation) % 5 = 0

AND EXTRACT(YEAR FROM CURRENT\_DATE) - year\_foundation > 0;

--7 самый молодой вокалист и какую группу он представляет

SELECT

m.member\_name,

m.surname,

mg.group\_name,

EXTRACT(YEAR FROM AGE(m.birth\_date)) AS age

FROM

member m

JOIN

role\_of\_member rom ON m.member\_id = rom.member\_id

JOIN

musical\_role mr ON rom.musical\_role\_id = mr.musical\_role\_id

JOIN

musical\_group mg ON rom.musical\_group\_id = mg.musical\_group\_id

WHERE

mr.role\_name = 'Вокалист'

ORDER BY

m.birth\_date DESC

LIMIT 1;

## **ПРИЛОЖЕНИЕ В**

**ССЫЛКА НА ИСХОДНЫЙ КОД В DB-FIDDLE**

<https://www.db-fiddle.com/f/ub22Sym6u2PhQcvazP3zfx/0>

## **ПРИЛОЖЕНИЕ С**

**ССЫЛКА НА PR**

<https://github.com/moevm/sql-2024-2384/pull/12>