

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 лабораторной 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 для проверки работоспособности
- Исходный код в виде .sql файла загрузить в виде PR в репо

Выполнение работы

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

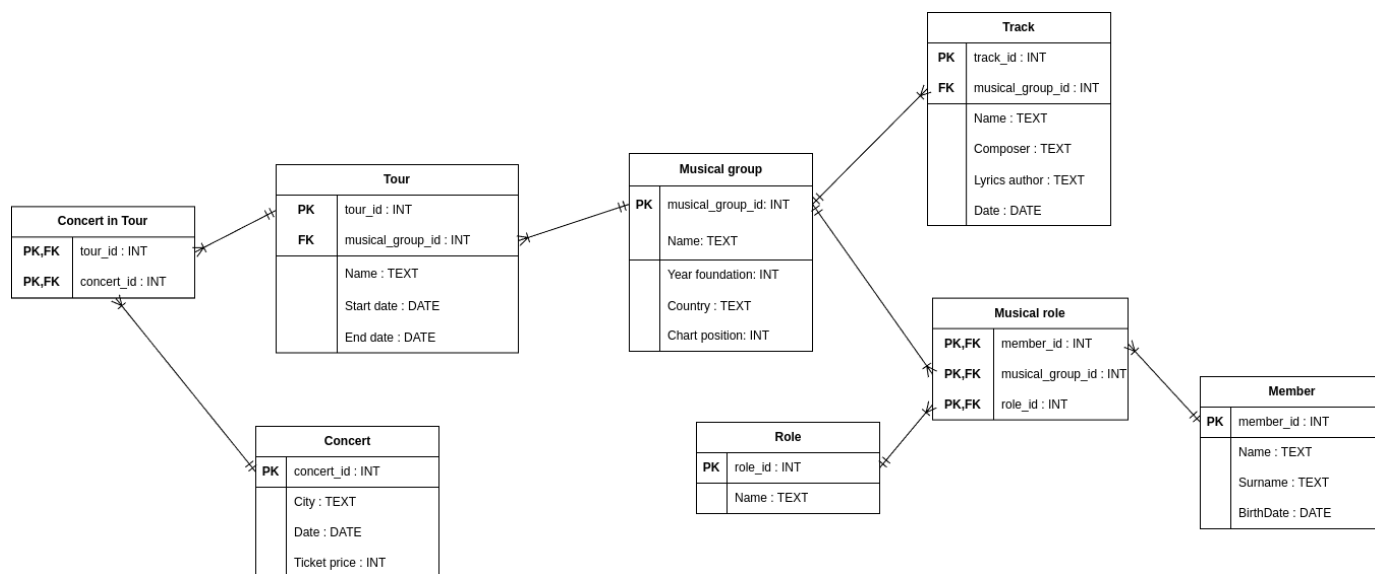


Рисунок 1 – структура БД

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.

```
mydb=> SELECT * FROM musical_group;
```

musical_group_id	group_name	year_foundation	country	chart_position
1	Queen	1970	Великобритания	5
2	The Cure	1978	Англия	2
3	Pink Floyd	1965	Англия	4
4	Монеточка	2015	Россия	3
5	ЛСП	2007	Беларусь	1
6	Pyrokinesis	2012	Россия	6

(6 rows)

Рисунок 2 – Таблица musical_group

```
mydb=> SELECT * FROM track;
```

track_id	musical_group_id	track_name	composer	lyrics_author	release_date
1	1	Bohemian Rhapsody	Queen	Фредди Меркьюри	1975-10-31
2	1	We Will Rock You	Брайан Мэй	Брайан Мэй	1977-10-07
3	2	Lovesong	Роберт Смит	Роберт Смит	1989-08-21
4	2	Friday I am in Love	Роберт Смит	Роберт Смит	1992-05-11
5	3	Another Brick in the Wall	Роджер Уотерс	Роджер Уотерс	1979-11-30
6	3	Time	Роджер Уотерс	Роджер Уотерс	1973-01-01
7	4	Это было в России	Витя Исаев	Витя Исаев	2024-05-01
8	4	Каждый раз	Елизавета Гырдымова	Елизавета Гырдымова	2018-05-25
9	5	Тело	Олег Савченко	Рома Англичанин	2017-04-28
10	5	Коктейль	Олег Савченко	Рома Англичанин	2013-05-13
11	6	Черное солнышко	Андрей Федорович	Андрей Федорович	2019-08-15
12	6	Дворцовый мост	Андрей Федорович	Виталий Клячин	2023-12-15

(12 rows)

Рисунок 3 – Таблица track

```
mydb=> SELECT * FROM member;
```

member_id	member_name	surname	birth_date
1	Брайн	Мэй	1947-07-19
2	Роджер	Тейлор	1949-07-26
3	Роберт	Смит	1959-04-21
4	Саймон	Гэллап	1960-06-01
5	Ник	Мейсон	1944-01-27
6	Роджер	Уотерс	1943-09-06
7	Елизавета	Гырдымова	1998-06-01
8	Олег	Савченко	1989-07-10
9	Петр	Клюев	1989-05-28
10	Андрей	Федорович	1995-12-16

(10 rows)

Рисунок 4 – Таблица member

```
mydb=> SELECT * FROM musical_role;
```

musical_role_id	role_name
1	Вокалист
2	Барабанщик
3	Гитарист
4	Бэк-вокалист
5	Автор песен
6	Бас-гитарист

(6 rows)

Рисунок 5 – Таблица musical_role

```

mydb=> SELECT * FROM role_of_member;
member_id | musical_group_id | musical_role_id
-----+-----+-----
          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
(17 rows)

```

Рисунок 6 – Таблица role_of_member

```

mydb=> SELECT * FROM tour;
tour_id | musical_group_id | tour_name | start_day | end_day
-----+-----+-----+-----+-----
        1 |                  1 | Magic Tour | 1986-06-07 | 1986-08-09
        2 |                  2 | Shows of a lost world | 2023-11-19 | 2023-12-10
        3 |                  3 | Syd BARRET | 2007-05-10 | 2007-05-10
        4 |                  4 | Молитвы. Анекдоты. Тосты | 2024-11-11 | 2024-11-30
        5 |                  5 | Бриллианты | 2024-10-12 | 2024-12-08
        6 |                  6 | Тур-2024 | 2024-09-06 | 2024-10-27
(6 rows)

```

Рисунок 7 – Таблица tour

```

mydb=> SELECT * FROM concert;
concert_id | city | date_concert | ticket_price
-----+-----+-----+-----
          1 | Париж | 1986-06-14 | 1000
          2 | Лондон | 1986-07-11 | 1000
          3 | Буэнос-Айрес | 2023-11-25 | 5000
          4 | Монтевидео | 2023-11-27 | 5100
          5 | Лондон | 2007-05-10 | 2000
          6 | Барселона | 2024-11-25 | 5800
          7 | Прага | 2024-11-30 | 7800
          8 | Тюмень | 2024-11-19 | 2600
          9 | Омск | 2024-12-08 | 3200
         10 | Орел | 2024-10-08 | 2500
         11 | Тамбов | 2024-10-11 | 2000
(11 rows)

```

Рисунок 8 – Таблица concert

```
mydb=> SELECT * FROM concert_in_tour;
tour_id | concert_id
-----+-----
        1 |          1
        1 |          2
        2 |          3
        2 |          4
        3 |          5
        4 |          6
        4 |          7
        5 |          8
        5 |          9
        6 |         10
        6 |         11
(11 rows)
```

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”.

```
mydb=> 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';
composer | lyrics_author | release_date | group_name
-----+-----+-----+-----
Андрей Федорович | Виталий Клячин | 2023-12-15 | Pyrokinesis
(1 row)
```

Рисунок 10 – Информация об определенной песне

- 2) Репертуар наиболее популярной группы?

```
mydb=> 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;
track_name | composer | lyrics_author | release_date
-----+-----+-----+-----
Тело | Олег Савченко | Рома Англичанин | 2017-04-28
Коктейль | Олег Савченко | Рома Англичанин | 2013-05-13
(2 rows)
```

Рисунок 11 – Репертуар самой популярной группы

3) Цена билета на последний концерт указанной группы?

Для теста была выбрана группа “Монеточка”.

```
mydb=> 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;  
ticket_price  
-----  
          7800  
(1 row)
```

Рисунок 12 – Цена билета

4) Состав исполнителей группы с заданным названием, их возраст и амплуа?

Для теста была выбрана группа “The Cure”.

```

mydb=> 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';

```

member_name	surname	age	role_name
Роберт	Смит	65	Вокалист
Роберт	Смит	65	Гитарист
Саймон	Гэллп	64	Бас-гитарист

(3 rows)

Рисунок 13 – Информация о составе определенной группы

5) Место и продолжительность гастролей группы с заданным названием?

Для теста была выбрана группа “ЛСП”.

```

mydb=> 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');

```

tour_name	city	duration
Бриллианты	Тюмень	57
Бриллианты	Омск	57

(2 rows)

Рисунок 14 – Информация о турах заданной группы

6) Какие группы в текущем году отмечают юбилей?


```

mydb=> 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;
group_name | year_foundation | anniversary
-----+-----+-----
(0 rows)

```

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:

```

mydb=> SELECT
    group_name,
    year_foundation,
    '2025' - year_foundation AS anniversary
FROM
    musical_group
WHERE
    ('2025' - year_foundation) % 5 = 0
    AND '2025' - year_foundation > 0;
group_name | year_foundation | anniversary
-----+-----+-----
Queen      | 1970            | 55
Pink Floyd | 1965            | 60
Монеточка  | 2015            | 10
(3 rows)

```

Рисунок 16 – Юбилей в конкретном году

- 7) Самый молодой вокалист? Какую группу он представляет?

```

mydb=> 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;
 member_name | surname | group_name | age
-----+-----+-----+---
 Елизавета  | Гырдымова | Монеточка  | 26
(1 row)

```

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', 'Гэллп', '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>