**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. 1304 |  | Байков Е.С. |
| Преподаватель |  | Заславский М.М. |

Санкт-Петербург

2023

## Цель работы.

Реализация базы данных в СУБД PostgreSQL.

## Задание.

Пусть требуется создать программную систему, предназначенную для работников библиотеки. Такая система должна обеспечивать хранение сведений об имеющихся в библиотеке книгах, о читателях библиотеки и читальных залах. Для каждой книги в БД должны храниться следующие сведения: название книги, автор (ы), издательство, год издания, число экземпляров этой книги в каждом зале библиотеки, а также шифр книги и дата закрепления книги за читателем. Сведения о читателях библиотеки должны включать номер читательского билета, фамилию читателя, номер паспорта, дату рождения, адрес, номер телефон, образование, наличие ученой степени. Читатели закрепляются за определенным залом и могут записываться и выписываться из библиотеки. Библиотека имеет несколько читальных залов, которые характеризуются номером, названием и вместимостью, то есть количеством людей, которые могут одновременно работать в зале. Библиотека может получать новые книги и списывать старые. Шифр книги может измениться в результате переклассификации, а номер читательского билета в результате перерегистрации. Библиотекарю могут потребоваться следующие сведения о текущем состоянии библиотеки:

* Какие книги закреплены за определенным читателем?
* Как называется книга с заданным шифром?
* Какой шифр у книги с заданным названием?
* Когда книга была закреплена за читателем?
* Кто из читателей взял книгу более месяца тому назад?
* За кем из читателей закреплены книги, количество экземпляров которых в библиотеке не превышает 2?
* Какое число читателей пользуется библиотекой?
* Сколько в библиотеке читателей младше 20 лет?

Необходимо развернуть PostgreSQL локально:

* 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 из предыдущей лабораторной работы
* Исходный код выложить на www.db-fiddle.com для проверки работоспособности
* Исходный код в виде .sql файла запушить в виде PR в репо
* В отчете описать:
* Цель
* Текст задания в соответствии с вариантом
* Скриншоты работы с СУБД PostgreSQL (psql / DBeaver / Datagrip, ...)
* Скриншоты на каждый запрос (или группу запросов) на изменение/таблицы с выводом результатов (ответ)
* Исходный код в приложении
* Ссылку на исходный код www.db-fiddle.com в приложении
* Ссылка на PR в приложении
* Вывод

## Выполнение работы.

1. Сперва была установлена СУБД PostgreSQL и pgAdmin4 на Windows. А также локально была создана база данных library (см. рис. 1).

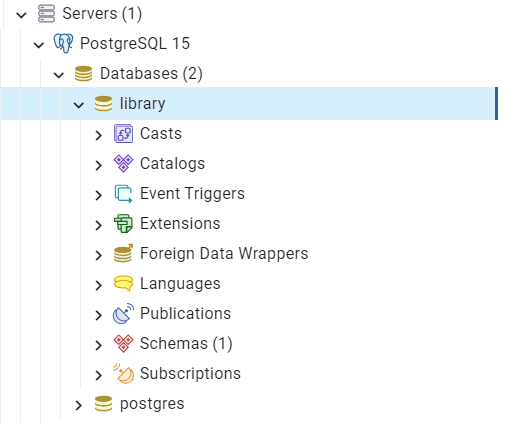


Рисунок 1 – База данных library.

1. Затем были созданы таблицы по структуре, которая была описана в предыдущей лабораторной работе. Запросы по созданию таблиц находятся в приложении А. Вид таблиц смотреть рисунки 2 – 10.



Рисунок 2 – Таблица book.



Рисунок 3 – Таблица author.



Рисунок 4 – Таблица publisher.

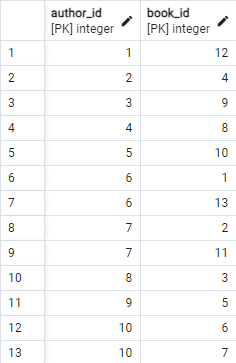


Рисунок 5 – Таблица author\_book.

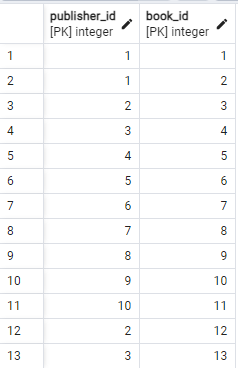


Рисунок 6 – Таблица publisher\_book.

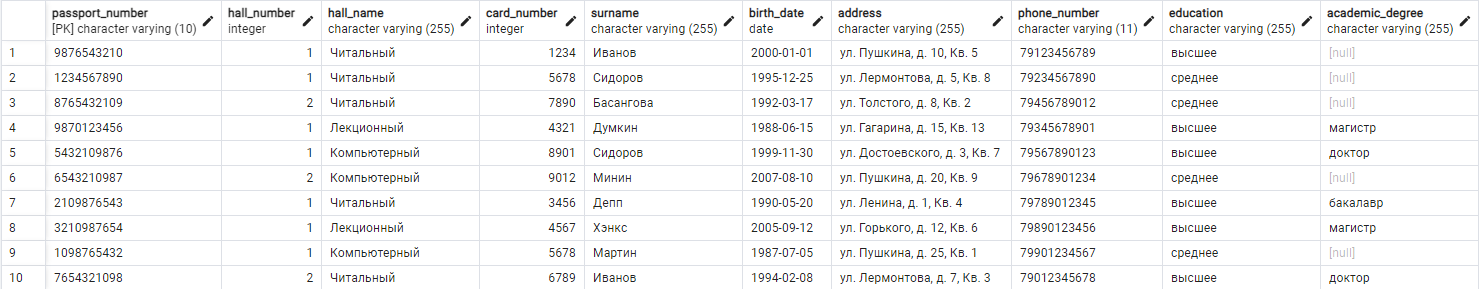


Рисунок 7 – Таблица library\_visitor.

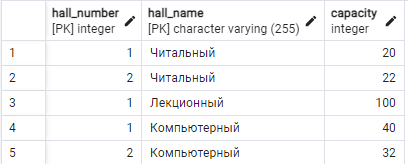


Рисунок 8 – Таблица reading\_hall.

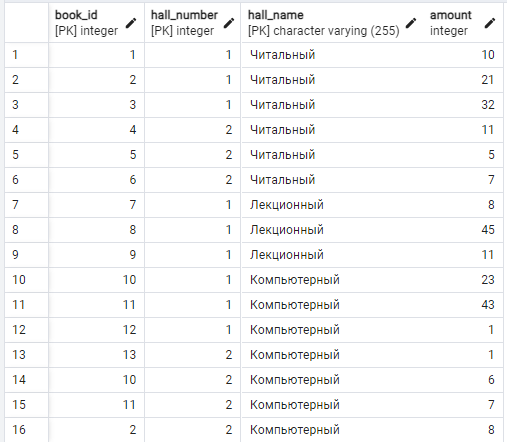


Рисунок 9 – Таблица reading\_hall\_has\_book.

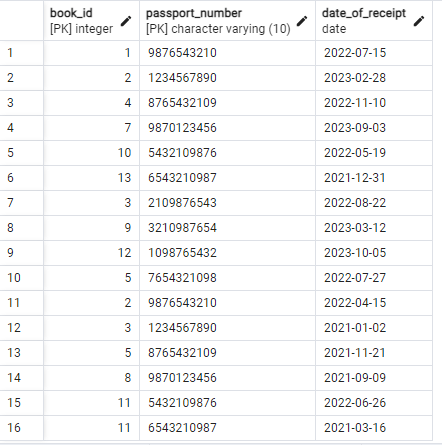


Рисунок 10 – Таблица visitor\_has\_book.

1. Были написаны запросы к таблицам, отвечающие вопросам в задании.

* Какие книги закреплены за определенным читателем?

Был написан запрос к базе данных (см. листинг 1)

Листинг 1.

SELECT book.title, library\_visitor.surname,

library\_visitor.passport\_number

FROM book

    INNER JOIN visitor\_has\_book USING(book\_id)

    INNER JOIN library\_visitor USING(passport\_number)

ORDER BY library\_visitor.surname, library\_visitor.passport\_number;

Результат изображён на рисунке 11.



Рисунок 11 – Результат запроса из листинга 1.

* Как называется книга с заданным шифром?

Был написан запрос к базе данных (см. листинг 2).

Листинг 2.

SELECT title, cipher FROM book

WHERE cipher = 99 OR cipher = 81;

Результат запроса изображён на рисунке 12.

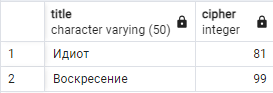


Рисунок 12 – результат выполнения запроса из листинга 2.

* Какой шифр у книги с заданным названием?

Был написан запрос к базе данных (см. листинг 3).

Листинг 3.

SELECT title, cipher FROM book

WHERE title = 'Реквием' OR title = 'Лолита';

Результат запроса изображён на рисунке 13.

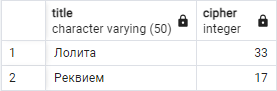


Рисунок 13 – результат выполнения запроса из листинга 3.

* Когда книга была закреплена за читателем?

Был написан запрос к базе данных (см. листинг 4).

Листинг 4.

SELECT book.title, library\_visitor.surname,

library\_visitor.passport\_number, visitor\_has\_book.date\_of\_receipt

FROM book

    INNER JOIN visitor\_has\_book USING(book\_id)

    INNER JOIN library\_visitor USING(passport\_number)

ORDER BY library\_visitor.surname, visitor\_has\_book.date\_of\_receipt;

Результат запроса изображён на рисунке 14.

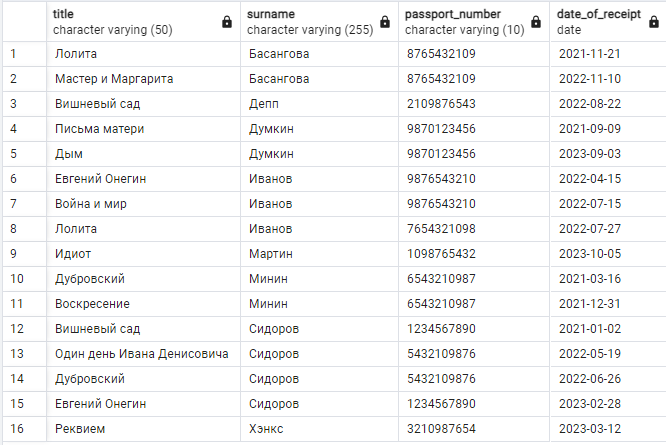


Рисунок 14 – результат выполнения запроса из листинга 4.

* Кто из читателей взял книгу более месяца тому назад?

Был написан запрос к базе данных (см. листинг 5).

Листинг 5.

SELECT library\_visitor.surname, library\_visitor.passport\_number

FROM library\_visitor

    INNER JOIN visitor\_has\_book USING(passport\_number)

WHERE visitor\_has\_book.date\_of\_receipt <= CURRENT\_DATE - INTERVAL '1 month'

GROUP BY (library\_visitor.surname, library\_visitor.passport\_number);

Результат запроса изображён на рисунке 15.

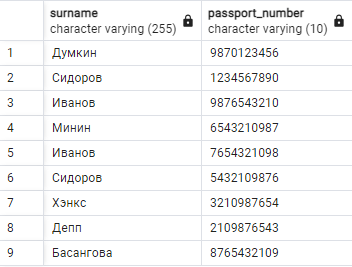


Рисунок 15 – результат выполнения запроса из листинга 5.

* За кем из читателей закреплены книги, количество экземпляров которых в библиотеке не превышает 2?

Был написан запрос к базе данных (см. листинг 6).

Листинг 6.

SELECT library\_visitor.surname, library\_visitor.passport\_number

FROM visitor\_has\_book

    INNER JOIN library\_visitor USING(passport\_number)

WHERE visitor\_has\_book.book\_id IN (

    SELECT reading\_hall\_has\_book.book\_id FROM reading\_hall\_has\_book

    GROUP BY reading\_hall\_has\_book.book\_id

    HAVING SUM(reading\_hall\_has\_book.amount) <= 2)

ORDER BY library\_visitor.surname;

Результат запроса изображён на рисунке 16.

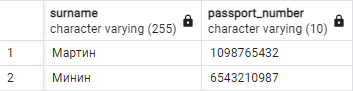


Рисунок 16 – результат выполнения запроса из листинга 6.

* Какое число читателей пользуется библиотекой?

Был написан запрос к базе данных (см. листинг 7).

Листинг 7.

SELECT COUNT(\*) AS "Количество читателей" FROM library\_visitor;

Результат запроса изображён на рисунке 17.

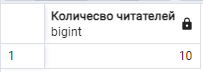


Рисунок 17 – результат выполнения запроса из листинга 7.

* Сколько в библиотеке читателей младше 20?

Был написан запрос к базе данных (см. листинг 8).

Листинг 8.

SELECT COUNT(\*) AS "Количество читателей младше 20" FROM library\_visitor

WHERE birth\_date > CURRENT\_DATE - INTERVAL '20 year';

Результат запроса изображён на рисунке 18.

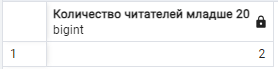


Рисунок 18 – результат выполнения запроса из листинга 8.

## Выводы.

В ходе выполнения работы, были получены навыки работы с СУБД PostgreSQL и pgAdmin4, а также была реализована база данных спроектированная в предыдущей работе.

# Приложение А ИСХОДНЫЙ КОД

Название файла: lab2.sql

CREATE TABLE book (

book\_id SERIAL PRIMARY KEY,

title VARCHAR(50) NOT NULL,

year\_of\_publication INT NOT NULL,

date\_of\_receipt DATE NOT NULL,

date\_of\_deletion DATE,

cipher INT NOT NULL

);

CREATE TABLE author (

author\_id SERIAL PRIMARY KEY,

author\_name VARCHAR(255)

);

CREATE TABLE publisher (

publisher\_id SERIAL PRIMARY KEY,

publisher\_name VARCHAR(255)

);

CREATE TABLE author\_book (

author\_id SERIAL,

book\_id SERIAL,

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

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

PRIMARY KEY (author\_id, book\_id)

);

CREATE TABLE publisher\_book (

publisher\_id SERIAL,

book\_id SERIAL,

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

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

PRIMARY KEY (publisher\_id, book\_id)

);

CREATE TABLE reading\_hall (

hall\_number INT NOT NULL,

hall\_name VARCHAR(255) NOT NULL,

PRIMARY KEY(hall\_number, hall\_name),

capacity INT NOT NULL

);

CREATE TABLE library\_visitor (

passport\_number VARCHAR(10) NOT NULL PRIMARY KEY,

hall\_number INT NOT NULL,

hall\_name VARCHAR(255) NOT NULL,

FOREIGN KEY (hall\_number, hall\_name) REFERENCES reading\_hall (hall\_number, hall\_name),

card\_number INT NOT NULL,

surname VARCHAR(255) NOT NULL,

birth\_date DATE NOT NULL,

address VARCHAR(255) NOT NULL,

phone\_number VARCHAR(11) NOT NULL,

education VARCHAR(255),

academic\_degree VARCHAR(255)

);

CREATE TABLE reading\_hall\_has\_book (

book\_id SERIAL,

hall\_number INT NOT NULL,

hall\_name VARCHAR(255) NOT NULL,

FOREIGN KEY (hall\_number, hall\_name) REFERENCES reading\_hall (hall\_number, hall\_name),

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

amount INT NOT NULL,

PRIMARY KEY (book\_id, hall\_number, hall\_name)

);

CREATE TABLE visitor\_has\_book (

book\_id SERIAL,

passport\_number VARCHAR(10) NOT NULL,

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

FOREIGN KEY (passport\_number) REFERENCES library\_visitor (passport\_number),

date\_of\_receipt DATE NOT NULL,

PRIMARY KEY (book\_id, passport\_number)

);

INSERT INTO author (author\_name)

VALUES

('Достоевский Ф.М.'),

('Булгаков М.А.'),

('Ахматова А.А.'),

('Есенин С.А.'),

('Солженицын А.И.'),

('Толстой Л.Н.'),

('Пушкин А.С.'),

('Чехов А.П.'),

('Набоков В.В.'),

('Тургенев И.С.');

INSERT INTO publisher (publisher\_name)

VALUES

('Эксмо'),

('АСТ'),

('Алгоритм'),

('Азбука'),

('Амфора'),

('АРМАДА'),

('РИПОЛ классик'),

('Книжный мир'),

('Манн, Иванов и Фербер'),

('Красная площадь');

INSERT INTO book (title, year\_of\_publication, date\_of\_receipt, date\_of\_deletion, cipher)

VALUES

('Война и мир', 1869, '2021-10-21', NULL, 10),

('Евгений Онегин', 1833, '2021-10-30', NULL, 32),

('Вишневый сад', 1903, '2021-11-21', '2023-01-21', 43),

('Мастер и Маргарита', 1966, '2012-09-21', '2023-01-22', 21),

('Лолита', 1958, '2021-07-18', NULL, 33),

('Отцы и дети', 1860, '2018-01-21', NULL, 73),

('Дым', 1867, '2019-12-13', NULL, 5),

('Письма матери', 1924, '2020-06-11', NULL, 11),

('Реквием', 1963, '2021-03-25', NULL, 17),

('Один день Ивана Денисовича', 1962, '2023-01-09', NULL, 22),

('Дубровский', 1833, '2022-12-22', NULL, 37),

('Идиот', 1869, '2020-04-14', NULL, 81),

('Воскресение', 1899, '2017-05-05', NULL, 99);

INSERT INTO publisher\_book (publisher\_id, book\_id)

VALUES

(1, 1),

(1, 2),

(2, 3),

(3, 4),

(4, 5),

(5, 6),

(6, 7),

(7, 8),

(8, 9),

(9, 10),

(10, 11),

(2, 12),

(3, 13);

INSERT INTO author\_book (author\_id, book\_id)

VALUES

(1, 12),

(2, 4),

(3, 9),

(4, 8),

(5, 10),

(6, 1),

(6, 13),

(7, 2),

(7, 11),

(8, 3),

(9, 5),

(10, 6),

(10, 7);

INSERT INTO reading\_hall (hall\_number, hall\_name, capacity)

VALUES

(1, 'Читальный', 20),

(2, 'Читальный', 22),

(1, 'Лекционный', 100),

(1, 'Компьютерный', 40),

(2, 'Компьютерный', 32);

INSERT INTO library\_visitor (passport\_number, hall\_number, hall\_name, card\_number, surname, birth\_date, address, phone\_number, education, academic\_degree)

VALUES

('9876543210', 1, 'Читальный', 1234, 'Иванов', '2000-01-01', 'ул. Пушкина, д. 10, Кв. 5', '79123456789', 'высшее', NULL),

('1234567890', 1, 'Читальный', 5678, 'Сидоров', '1995-12-25', 'ул. Лермонтова, д. 5, Кв. 8', '79234567890', 'среднее', NULL),

('8765432109', 2, 'Читальный', 7890, 'Басангова', '1992-03-17', 'ул. Толстого, д. 8, Кв. 2', '79456789012', 'среднее', NULL),

('9870123456', 1, 'Лекционный', 4321, 'Думкин', '1988-06-15', 'ул. Гагарина, д. 15, Кв. 13', '79345678901', 'высшее', 'магистр'),

('5432109876', 1, 'Компьютерный', 8901, 'Сидоров', '1999-11-30', 'ул. Достоевского, д. 3, Кв. 7', '79567890123', 'высшее', 'доктор'),

('6543210987', 2, 'Компьютерный', 9012, 'Минин', '2007-08-10', 'ул. Пушкина, д. 20, Кв. 9', '79678901234', 'среднее', NULL),

('2109876543', 1, 'Читальный', 3456, 'Депп', '1990-05-20', 'ул. Ленина, д. 1, Кв. 4', '79789012345', 'высшее', 'бакалавр'),

('3210987654', 1, 'Лекционный', 4567, 'Хэнкс', '2005-09-12', 'ул. Горького, д. 12, Кв. 6', '79890123456', 'высшее', 'магистр'),

('1098765432', 1, 'Компьютерный', 5678, 'Мартин', '1987-07-05', 'ул. Пушкина, д. 25, Кв. 1', '79901234567', 'среднее', NULL),

('7654321098', 2, 'Читальный', 6789, 'Иванов', '1994-02-08', 'ул. Лермонтова, д. 7, Кв. 3', '79012345678', 'высшее', 'доктор');

INSERT INTO reading\_hall\_has\_book (book\_id, hall\_number, hall\_name, amount)

VALUES

(1, 1, 'Читальный', 10),

(2, 1, 'Читальный', 21),

(3, 1, 'Читальный', 32),

(4, 2, 'Читальный', 11),

(5, 2, 'Читальный', 5),

(6, 2, 'Читальный', 7),

(7, 1, 'Лекционный', 8),

(8, 1, 'Лекционный', 45),

(9, 1, 'Лекционный', 11),

(10, 1, 'Компьютерный', 23),

(11, 1, 'Компьютерный', 43),

(12, 1, 'Компьютерный', 1),

(13, 2, 'Компьютерный', 1),

(10, 2, 'Компьютерный', 6),

(11, 2, 'Компьютерный', 7),

(2, 2, 'Компьютерный', 8);

INSERT INTO visitor\_has\_book (book\_id, passport\_number, date\_of\_receipt)

VALUES

(1, '9876543210', '2022-07-15'),

(2, '1234567890', '2023-02-28'),

(4, '8765432109', '2022-11-10'),

(7, '9870123456', '2023-09-03'),

(10, '5432109876', '2022-05-19'),

(13, '6543210987', '2021-12-31'),

(3, '2109876543', '2022-08-22'),

(9, '3210987654', '2023-03-12'),

(12, '1098765432', '2023-10-05'),

(5, '7654321098', '2022-07-27'),

(2, '9876543210', '2022-04-15'),

(3, '1234567890', '2021-01-02'),

(5, '8765432109', '2021-11-21'),

(8, '9870123456', '2021-09-09'),

(11, '5432109876', '2022-06-26'),

(11, '6543210987', '2021-03-16');

SELECT book.title, library\_visitor.surname, library\_visitor.passport\_number

FROM book

INNER JOIN visitor\_has\_book USING(book\_id)

INNER JOIN library\_visitor USING(passport\_number)

ORDER BY library\_visitor.surname, library\_visitor.passport\_number;

SELECT title, cipher FROM book

WHERE cipher = 99 OR cipher = 81;

SELECT title, cipher FROM book

WHERE title = 'Реквием' OR title = 'Лолита';

SELECT book.title, library\_visitor.surname, library\_visitor.passport\_number, visitor\_has\_book.date\_of\_receipt

FROM book

INNER JOIN visitor\_has\_book USING(book\_id)

INNER JOIN library\_visitor USING(passport\_number)

ORDER BY library\_visitor.surname, visitor\_has\_book.date\_of\_receipt;

SELECT library\_visitor.surname, library\_visitor.passport\_number

FROM library\_visitor

INNER JOIN visitor\_has\_book USING(passport\_number)

WHERE visitor\_has\_book.date\_of\_receipt <= CURRENT\_DATE - INTERVAL '1 month'

GROUP BY (library\_visitor.surname, library\_visitor.passport\_number);

SELECT library\_visitor.surname, library\_visitor.passport\_number

FROM visitor\_has\_book

INNER JOIN library\_visitor USING(passport\_number)

WHERE visitor\_has\_book.book\_id IN (

SELECT reading\_hall\_has\_book.book\_id FROM reading\_hall\_has\_book

GROUP BY reading\_hall\_has\_book.book\_id

HAVING SUM(reading\_hall\_has\_book.amount) <= 2)

ORDER BY library\_visitor.surname;

SELECT COUNT(\*) AS "Количесво читателей" FROM library\_visitor;

SELECT COUNT(\*) AS "Количество читателей младше 20" FROM library\_visitor

WHERE birth\_date > CURRENT\_DATE - INTERVAL '20 year';

# Приложение Б ССылки

https://github.com/moevm/sql-2023-1304/pull/34

https://www.db-fiddle.com/f/uGUvp9b7SxcihCwQqfPkyz/11