

Лабораторная работа №9 Подзапросы в SQL

Цель работы:

Освоить использование подзапросов в SQL: в предложении WHERE с операторами сравнения, IN, EXISTS. Научиться комбинировать простые запросы для решения сложных задач.

Программное обеспечение: PostgreSQL, DBeaver

Время: 2 академических часа

Подготовка базы данных

Создание таблиц для библиотечной системы

```
SET search_path TO ваш_логин;

-- Авторы
CREATE TABLE authors (
    author_id SERIAL PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    birth_year INTEGER,
    country VARCHAR(50)
);

-- Книги
CREATE TABLE books (
    book_id SERIAL PRIMARY KEY,
    title VARCHAR(200) NOT NULL,
    author_id INTEGER REFERENCES authors(author_id),
    publication_year INTEGER,
    genre VARCHAR(50),
    pages INTEGER,
    price DECIMAL(8,2)
);
```

```
-- Читатели
CREATE TABLE readers (
    reader_id SERIAL PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    email VARCHAR(100) UNIQUE,
    registration_date DATE DEFAULT CURRENT_DATE,
    birth_date DATE
);

-- Выдачи книг
CREATE TABLE book_loans (
    loan_id SERIAL PRIMARY KEY,
    book_id INTEGER REFERENCES books(book_id),
    reader_id INTEGER REFERENCES readers(reader_id),
    loan_date DATE NOT NULL,
    return_date DATE,
    due_date DATE NOT NULL
);
```

Заполнение таблиц тестовыми данными

```
-- Авторы
INSERT INTO authors (first_name, last_name, birth_year, country) VALUES
('Лев', 'Толстой', 1828, 'Россия'),
('Федор', 'Достоевский', 1821, 'Россия'),
('Антон', 'Чехов', 1860, 'Россия'),
('Александр', 'Пушкин', 1799, 'Россия'),
('Михаил', 'Лермонтов', 1814, 'Россия'),
('Эрнест', 'Хемингуэй', 1899, 'США'),
('Агата', 'Кристи', 1890, 'Великобритания'),
('Джордж', 'Оруэлл', 1903, 'Великобритания'),
('Джоан', 'Роулинг', 1965, 'Великобритания'),
('Стивен', 'Кинг', 1947, 'США');

-- Книги
INSERT INTO books (title, author_id, publication_year, genre, pages, price) VALUES
('Война и мир', 1, 1869, 'Роман', 1225, 1500.00),
('Анна Каренина', 1, 1877, 'Роман', 864, 1200.00),
('Преступление и наказание', 2, 1866, 'Роман', 671, 950.00),
```

```
('Идиот', 2, 1869, 'Роман', 640, 900.00),
('Вишневый сад', 3, 1904, 'Пьеса', 96, 500.00),
('Евгений Онегин', 4, 1833, 'Роман в стихах', 240, 800.00),
('Герой нашего времени', 5, 1840, 'Роман', 320, 750.00),
('Прощай, оружие!', 6, 1929, 'Роман', 355, 850.00),
('Убийство в Восточном экспрессе', 7, 1934, 'Детектив', 256, 700.00),
('1984', 8, 1949, 'Антиутопия', 328, 900.00),
('Гарри Поттер и философский камень', 9, 1997, 'Фэнтези', 320, 1100.00),
('Сияние', 10, 1977, 'Ужасы', 447, 950.00),
('Капитанская дочка', 4, 1836, 'Повесть', 320, 650.00),
('Братья Карамазовы', 2, 1880, 'Роман', 824, 1300.00),
('Старик и море', 6, 1952, 'Повесть', 128, 600.00);
```

-- Читатели

```
INSERT INTO readers (first_name, last_name, email, registration_date,
birth_date) VALUES
('Иван', 'Иванов', 'ivanov@mail.ru', '2022-01-15', '1995-03-20'),
('Петр', 'Петров', 'petrov@gmail.com', '2022-02-10', '1998-07-12'),
('Мария', 'Сидорова', 'sidorova@yandex.ru', '2022-03-05', '1993-11-30'),
('Анна', 'Козлова', 'kozlova@mail.ru', '2022-04-20', '2000-01-25'),
('Сергей', 'Смирнов', 'smirnov@gmail.com', '2022-05-18', '1991-09-14'),
('Елена', 'Попова', 'popova@yandex.ru', '2022-06-22', '1996-12-08'),
('Алексей', 'Васильев', 'vasiliev@mail.ru', '2022-07-30', '1999-04-17'),
('Ольга', 'Михайлова', 'mikhailova@gmail.com', '2022-08-14', '1994-06-03'),
('Дмитрий', 'Федоров', 'fedorov@yandex.ru', '2022-09-09', '1992-08-28'),
('Наталья', 'Морозова', 'morozova@mail.ru', '2022-10-25', '1997-05-19');
```

-- Выдачи книг

```
INSERT INTO book_loans (book_id, reader_id, loan_date, return_date,
due_date) VALUES
(1, 1, '2024-01-10', '2024-01-25', '2024-02-10'),
(3, 1, '2024-02-05', '2024-02-20', '2024-02-25'),
(5, 2, '2024-01-15', NULL, '2024-02-15'),
(7, 2, '2024-02-01', '2024-02-10', '2024-02-15'),
(9, 3, '2024-01-20', '2024-02-05', '2024-02-05'),
(11, 3, '2024-02-10', NULL, '2024-03-10'),
(2, 4, '2024-01-25', '2024-02-15', '2024-02-25'),
(4, 5, '2024-02-03', NULL, '2024-03-03'),
(6, 6, '2024-01-30', '2024-02-12', '2024-02-20'),
(8, 7, '2024-02-08', NULL, '2024-03-08'),
```

```
(10, 8, '2024-01-12', '2024-01-30', '2024-02-01'),  
(12, 9, '2024-02-15', NULL, '2024-03-15'),  
(13, 10, '2024-01-18', '2024-02-01', '2024-02-08'),  
(14, 1, '2024-02-20', NULL, '2024-03-20'),  
(15, 2, '2024-01-22', '2024-02-10', '2024-02-10');
```

Примеры подзапросов

1. Подзапросы в WHERE с операторами сравнения

-- Найти книги, которые дороже средней цены

```
SELECT title, price  
FROM books  
WHERE price > (SELECT AVG(price) FROM books);
```

-- Найти самого молодого автора

```
SELECT first_name, last_name, birth_year  
FROM authors  
WHERE birth_year = (SELECT MAX(birth_year) FROM authors);
```

-- Найти книги, изданные в тот же год, что и "Война и мир"

```
SELECT title, publication_year  
FROM books  
WHERE publication_year = (  
    SELECT publication_year  
    FROM books  
    WHERE title = 'Война и мир'  
);
```

2. Подзапросы в WHERE с оператором IN

-- Найти книги авторов из России

```
SELECT title, publication_year  
FROM books  
WHERE author_id IN (  
    SELECT author_id  
    FROM authors  
    WHERE country = 'Россия'  
);
```

```
-- Найти читателей, которые брали книги в январе 2024
SELECT first_name, last_name
FROM readers
WHERE reader_id IN (
    SELECT reader_id
    FROM book_loans
    WHERE EXTRACT(MONTH FROM loan_date) = 1
    AND EXTRACT(YEAR FROM loan_date) = 2024
);
```

```
-- Найти книги, которые еще не возвращены
SELECT title
FROM books
WHERE book_id IN (
    SELECT book_id
    FROM book_loans
    WHERE return_date IS NULL
);
```

3. Подзапросы в WHERE с оператором EXISTS

```
-- Найти авторов, у которых есть книги дороже 1000 рублей
SELECT first_name, last_name
FROM authors a
WHERE EXISTS (
    SELECT 1
    FROM books b
    WHERE b.author_id = a.author_id
    AND b.price > 1000
);
```

```
-- Найти читателей, которые никогда не брали книги
SELECT first_name, last_name
FROM readers r
WHERE NOT EXISTS (
    SELECT 1
    FROM book_loans bl
    WHERE bl.reader_id = r.reader_id
);
```

```
-- Найти книги, которые были взяты более одного раза
SELECT title
FROM books b
WHERE EXISTS (
    SELECT 1
    FROM book_loans bl
    WHERE bl.book_id = b.book_id
    GROUP BY bl.book_id
    HAVING COUNT(*) > 1
);
```

4. Подзапросы в SELECT

```
-- Для каждой книги показать автора
SELECT
    title,
    (SELECT CONCAT(first_name, ' ', last_name)
     FROM authors a
     WHERE a.author_id = b.author_id) as author_name
FROM books b;

-- Для каждой выдачи показать название книги и читателя
SELECT
    loan_date,
    (SELECT title FROM books WHERE book_id = bl.book_id) as book_title,
    (SELECT CONCAT(first_name, ' ', last_name)
     FROM readers
     WHERE reader_id = bl.reader_id) as reader_name
FROM book_loans bl;
```

5. Подзапросы в FROM

```
-- Найти книги, цена которых выше средней по жанру
SELECT b.title, b.genre, b.price, avg_price.avg_genre_price
FROM books b
JOIN (
    SELECT genre, AVG(price) as avg_genre_price
    FROM books
    GROUP BY genre
) avg_price ON b.genre = avg_price.genre
WHERE b.price > avg_price.avg_genre_price;
```

```
-- Найти авторов и количество их книг
SELECT a.first_name, a.last_name, book_count.count
FROM authors a
JOIN (
    SELECT author_id, COUNT(*) as count
    FROM books
    GROUP BY author_id
) book_count ON a.author_id = book_count.author_id;
```

Варианты заданий для студентов

Вариант 1

Подзапросы с операторами сравнения (=, >, <):

- Найти все книги, изданные позже самого раннего произведения Агаты Кристи
- Найти авторов, которые родились раньше Льва Толстого
- Найти книги, которые дешевле самой дорогой книги в жанре "Роман"

Подзапросы с IN:

- Найти читателей, которые брали книги русских авторов
- Найти книги, которые брал читатель с email 'ivanov@mail.ru'
- Найти авторов, чьи книги были взяты в феврале 2024

Подзапросы с JOIN:

- Показать все выдачи с названиями книг и именами читателей
- Найти книги и их авторов для произведений, взятых в январе 2024
- Показать читателей и книги, которые они еще не вернули

Вариант 2

Подзапросы с операторами сравнения:

- Найти книги, которые толще средней длины книги
- Найти авторов, которые моложе Эрнеста Хемингуэя
- Найти книги, изданные в тот же год, что и "1984"

Подзапросы с IN:

- Найти книги в жанрах, к которым относятся произведения Достоевского
- Найти читателей, которые брали книги дороже 1000 рублей
- Найти книги, которые были взяты читателями, зарегистрированными в 2022 году

Подзапросы с JOIN:

7. Показать все книги с информацией об авторе (имя, страна)
8. Найти просроченные выдачи (книги не возвращены после due_date)
9. Показать читателей и количество взятых ими книг

Вариант 3

Подзапросы с операторами сравнения:

1. Найти авторов, у которых больше книг, чем у Пушкина
2. Найти книги, которые короче самой длинной книги Толстого
3. Найти читателей, которые зарегистрировались раньше самого молодого читателя

Подзапросы с IN:

4. Найти книги, которые никогда не брали
5. Найти читателей, которые брали книги британских авторов
6. Найти авторов, чьи книги были взяты читателем Петровым

Подзапросы с JOIN:

7. Показать историю выдач с деталями книги и читателя
8. Найти книги, которые в данный момент находятся на руках
9. Показать авторов и их самые дорогие книги

Вариант 4

Подзапросы с операторами сравнения:

1. Найти книги, которые дороже средней цены книг того же автора
2. Найти авторов, которые старше среднего возраста всех авторов
3. Найти читателей, которые моложе самого старшего читателя, бравшего книги

Подзапросы с IN:

4. Найти книги в жанрах, представленных в библиотеке
5. Найти читателей, которые брали книги Толстого или Достоевского
6. Найти книги, изданные в годы, когда публиковались русские авторы

Подзапросы с JOIN:

7. Показать все выдачи с просрочкой (текущая дата > due_date и return_date IS NULL)
8. Найти книги и их авторов для определенного жанра
9. Показать читателей и даты их первой выдачи

Вариант 5

Подзапросы с операторами сравнения:

1. Найти книги, которые были взяты позже самой ранней выдачи
2. Найти авторов, у которых меньше книг, чем у Кинга
3. Найти читателей, которые зарегистрировались позже среднего срока регистрации

Подзапросы с IN:

4. Найти книги, которые были возвращены вовремя (до due_date)
5. Найти читателей, которые брали книги в определенный период
6. Найти авторов, чьи книги есть в жанре "Роман" или "Детектив"

Подзапросы с JOIN:

7. Показать книги с информацией о том, сколько раз их брали
 8. Найти читателей и книги, которые они взяли в последний раз
 9. Показать авторов и годы издания их первой книги
-

Дополнительные задания для студентов

Задание 1: Комбинированные запросы

1. Используя JOIN и подзапрос, найдите книги и их авторов, где книги были взяты более одного раза
2. Используя подзапрос в FROM, создайте список авторов с количеством их книг в библиотеке
3. Используя коррелированный подзапрос, для каждого читателя найдите самую раннюю дату выдачи

Задание 2: Аналитические запросы

1. Найдите "популярных" авторов (тех, чьи книги брали хотя бы один раз)
 2. Найдите "активных" читателей (тех, кто брал книги разных авторов)
 3. Создайте отчет: книга, автор, был ли экземпляр взят хотя бы раз
-

Требования к отчету

1. **SQL-запросы** для каждого задания с комментариями
 2. **Результаты выполнения** (первые 10 строк каждого запроса)
 3. **Объяснение логики** работы каждого подзапроса
 4. **Сравнение** разных подходов к решению одних и тех же задач
-

Контрольные вопросы

1. В чем разница между подзапросом и JOIN? Когда что использовать?
2. Что такое коррелированный подзапрос и как он работает?
3. В чем разница между IN, EXISTS и JOIN при фильтрации?
4. Какие ограничения есть у подзапросов в разных частях SQL-запроса?
5. Как оптимизировать запросы с подзапросами?
6. Можно ли использовать подзапрос в ORDER BY? Если да, то как?
7. Какие проблемы производительности могут возникнуть при использовании подзапросов?