**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра систем автоматизированного проектирования (САПР)**

**отчет**

**по лабораторной работе № 5**

**по дисциплине «Базы данных»**

Тема: **«Работа с вложенными запросами»**

|  |  |  |
| --- | --- | --- |
| Студенты гр. 3311 | Аршин А. Д  Баймухамедов Р. Р.  Пасечный Л. В. |  |
| Преподаватель | Новакова Н. Е. |  |

Санкт-Петербург

2025

**Цель работы**

Научиться писать и применять вложенные запросы. В лабораторной работе используется БД Library.

**Упражнение 1**

Выполним запрос, использующий производные таблицы. Запрос, формирующий производную таблицу, будет возвращать столбец juveline.adult\_member\_no и количество подростков для каждого взрослого читателя библиотеки, имеющего более трех детей (подростков) записанных в библиотеку (являющихся ее читателями). Список раздела SELECT основного запроса будет включать поля adult\_member\_no и No\_Of\_Children из производного запроса и поле expr\_date из таблицы adult. Выполним запрос, использующий производную таблицу

*Запрос*

SELECT jv.adult\_member\_no, No\_Of\_Children, ad.expr\_date

FROM adult AS ad

JOIN

(

SELECT adult\_member\_no, COUNT(member\_no) AS 'No\_Of\_Children'

FROM juvenile

GROUP BY adult\_member\_no

HAVING COUNT(member\_no)>3

) AS jv

ON ad.member\_no = jv.adult\_member\_no

*Результат*

adult\_member\_no No\_Of\_Children expr\_date

1 4 2006-03-19 21:32:38.513

3 4 2006-03-21 21:32:38.513

5 4 2006-03-23 21:32:38.513

7 4 2006-03-25 21:32:38.513

9 4 2006-03-27 21:32:38.513…

… … …

(Затронуто строк: 248)

Время выполнения: 2025-10-08T20:30:27.1595373+03:00

Создадим запрос с производной таблицей в виде двух разных запросов

*Запрос*

WITH adult\_w\_kids AS(

SELECT jv.adult\_member\_no, COUNT(jv.member\_no) AS 'count\_of\_child', ad.expr\_date

FROM adult AS ad

JOIN

juvenile AS jv

ON ad.member\_no = jv.adult\_member\_no

GROUP BY jv.adult\_member\_no, ad.expr\_date

HAVING COUNT(jv.member\_no)>3

)

SELECT adult\_member\_no, count\_of\_child, expr\_date

FROM adult\_w\_kids

*Результат*

adult\_member\_no count\_of\_child expr\_date

1 4 2006-03-19 21:32:38.513

3 4 2006-03-21 21:32:38.513

5 4 2006-03-23 21:32:38.513

7 4 2006-03-25 21:32:38.513

9 4 2006-03-27 21:32:38.513

… … …

(Затронуто строк: 248)

Время выполнения: 2025-10-08T21:19:07.4869212+03:00

**Упражнение 2**

Использование подзапроса как части условия поиска

Напишем запрос, который соединяет таблицы member и loanhist и возвращает значения firstname, lastname, isbn и fine\_paid для всех строк. Используем этот запрос как критерий выбора в предложении WHERE, так чтобы возвращались только те записи, в которых штраф имеет максимальное значение. Включим ключевое слово DISTINCT.

*Запрос*

WITH who\_what\_paid AS (

SELECT mb.firstname, mb.lastname, lh.isbn, lh.fine\_paid

FROM

member AS mb

JOIN

loanhist AS lh

ON mb.member\_no = lh.member\_no

)

SELECT DISTINCT firstname, lastname, isbn, fine\_paid

FROM who\_what\_paid

WHERE fine\_paid = (

SELECT MAX(fine\_paid)

FROM who\_what\_paid

)

*Результат*

firstname lastname isbn fine\_paid

Angela Hightower 221 8,00

Clair Rothenberg 32 8,00

Karl Wolfe-Hellene 403 8,00

William Erickson 30 8,00

(Затронуто строк: 4)

Время выполнения: 2025-10-08T21:11:57.8384799+03:00

Теперь напишем запрос к таблице title, loan и reservation, возвращающий значения 4 полей: title\_no, title, isbn и Total Reserved. Поле Total Reserved представляет собой количество резервных экземпляров для каждой книги. Нужно отобразить те записи, которых в резерве или более 50, или менее 5.

*Запрос*

SELECT tt.title\_no, tt.title, ln.isbn, COUNT(\*) AS 'Total\_Reserved'

FROM title AS tt

JOIN loan AS ln

ON tt.title\_no = ln.title\_no

WHERE ln.isbn IN (

SELECT isbn

FROM reservation

GROUP BY isbn

HAVING COUNT(\*)>50 OR COUNT(\*)<5

)

GROUP BY tt.title\_no, tt.title, ln.isbn

*Результат*

title\_no title isbn Total\_Reserved

1 Last of the Mohicans 1 2

5 Fall of the House of Usher 43 2

25 The Black Tulip 246 2

29 Misalliance 288 2

33 The First 100,000 Prime Numbers 330 2

… … … …

(Затронуто строк: 11)

Время выполнения: 2025-10-08T21:59:49.7759635+03:00

**Упражнение 3**

Используем коррелированные подзапросы. В этом задании создадим запрос, использующий коррелированный подзапрос для вычисления значений, основанных на данных из внешнего запроса, и использующий эти значения как часть условия сравнения. Необходимо отобразить список читателей, имеющих сумму штрафов, превышающую 5 у.е.

*Запрос*

SELECT mb.member\_no, mb.lastname

FROM member AS mb

WHERE (

SELECT SUM(fine\_assessed) AS 'total\_fine'

FROM loanhist AS lh

WHERE lh.member\_no = mb.member\_no

) > 5

*Результат*

member\_no lastname

82 Chen

259 Hightower

322 LaBrie

502 Rudd

731 Brooke

… …

Внимание! Значение NULL исключено в агрегатных или других операциях SET.

(Затронуто строк: 57)

Время выполнения: 2025-10-08T22:21:39.5139467+03:00

**Вывод**

В рамках лабораторной работы закреплены базовые принципы использования вложенных запросов в SQL. На практике рассмотрены и применены разные типы подзапросов:

1. Производные таблицы. Освоено построение временных наборов данных и их соединение через `JOIN`, что позволяет структурировать сложные запросы и повышать их читаемость.
2. Подзапросы-выражения. Изучено применение скалярных подзапросов для получения единственного значения и последующей фильтрации данных в условиях `WHERE`.
3. Подзапросы со списками значений. Отработано использование оператора `IN` для работы с наборами ключей, формируемых внутренними выборками.