**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

отчет

**по лабораторной работе №5**

**по дисциплине «Базы данных»**

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

|  |  |  |
| --- | --- | --- |
| Студент гр. 1335 |  | Максимов Ю.Е. |
| Преподаватель |  | Новакова Н.Е. |

Санкт-Петербург

2024

Цель работы: научиться писать вложенные запросы.

Упражнение 1 – использование вложенных запросов как производных таблиц.

1. **Задание 1.1**

* **Запрос**

use Library;

SELECT a.member\_no, j.No\_Of\_Children,CONVERT(NVARCHAR(10), a.expr\_date, 104) expr\_date

FROM dbo.adult a

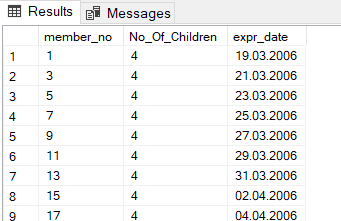
OUTER APPLY(SELECT COUNT(j.member\_no) No\_Of\_Children

FROM dbo.juvenile j

WHERE j.adult\_member\_no = a.member\_no)j

WHERE j.No\_Of\_Children > 3

* **Ответ**



(248 rows affected)

Completion time: 2024-10-15T00:25:28.443533+03:00

1. **Задание 1.2**

* **Запрос**

use Library;

SELECT a.member\_no adult\_member\_no, COUNT(j.member\_no) No\_Of\_Children, CONVERT(NVARCHAR(10), a.expr\_date, 104) expr\_date

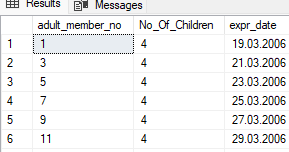
FROM dbo.juvenile j

INNER JOIN dbo.adult a ON a.member\_no = j.adult\_member\_no

GROUP BY a.expr\_date, a.member\_no

HAVING COUNT(j.member\_no) > 3

* **Ответ**



(248 rows affected)

Completion time: 2024-10-15T01:26:30.443533+03:00

Упражнение 2 – использование подзапросов, как выражений:

1. **Задание 2.1**

* **Запрос**

DECLARE @max\_paid DECIMAL(18, 2) = (SELECT MAX(l.fine\_paid) FROM dbo.loanhist l)

SELECT m.firstname member\_firstname, m.lastname member\_lastname, l.isbn loanhist\_isbn, l.fine\_paid loanhist\_fine\_paidMax

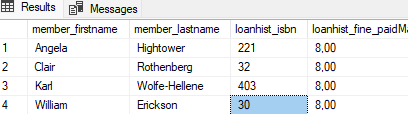
FROM dbo.member m

INNER JOIN dbo.loanhist l ON l.member\_no = m.member\_no

WHERE l.fine\_paid = @max\_paid

GROUP BY m.firstname, m.lastname, l.isbn, l.fine\_paid

* **Ответ**



Completion time: 2024-10-15T02:27:32.443533+03:00

1. **Задание 2.2**

* **Запрос**

SELECT DISTINCT m.firstname member\_firstname, m.lastname member\_lastname, l.isbn loanhist\_isbn, l.fine\_paid loanhist\_fine\_paidMax

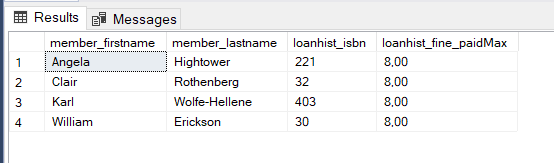
FROM dbo.member m

INNER JOIN dbo.loanhist l ON l.member\_no = m.member\_no

WHERE (SELECT MAX(lt.fine\_paid) FROM dbo.loanhist lt) = l.fine\_paid

ORDER BY loanhist\_fine\_paidMax DESC

* **Ответ**



Completion time: 2024-10-15T03:28:34.443533+03:00

1. **Задание 2.3**

SELECT t.title\_no, t.title, l.isbn, COUNT(r.isbn) cnt

INTO #temp

FROM dbo.loan l

INNER JOIN dbo.title t ON t.title\_no = l.title\_no

INNER JOIN dbo.reservation r ON r.member\_no = l.member\_no

GROUP BY t.title\_no, t.title, l.isbn

HAVING COUNT(r.isbn) > 50 OR COUNT(r.isbn) < 5

* **Ответ**

(117 rows affected)

Completion time: 2024-11-16T09:31:46.8209180+03:00

1. **Задание 2.4**

* **Запрос**

SELECT t.title\_no, t.title, l.isbn, COUNT(r.isbn)

FROM dbo.loan l

INNER JOIN dbo.title t ON t.title\_no = l.title\_no

INNER JOIN dbo.reservation r ON r.member\_no = l.member\_no

WHERE t.title\_no IN(SELECT title\_no FROM #temp)

GROUP BY t.title\_no, t.title, l.isbn

HAVING COUNT(r.isbn) < 5**Ответ**



(117 rows affected)

Упражнение 3 – использование коррелированных подзапросов:

1. **Задание**

* **Запрос**

SELECT m.member\_no,

m.lastname,

l.paid

FROM dbo.member m

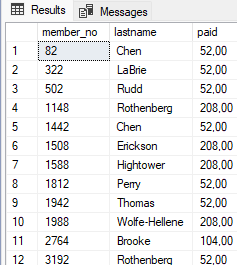
OUTER APPLY(SELECT SUM(ISNULL(l.fine\_paid, '0.00')) paid

FROM dbo.loanhist l

WHERE l.member\_no = m.member\_no)l

WHERE l.paid > '5.00'

* **Ответ**



(25 rows affected)

Completion time: 2024-10-15T04:29:36.443533+03:00

**Вывод**

В ходе выполнения лабораторной работы я научился создавать и использовать вложенные запросы (подзапросы) в языке SQL для решения сложных задач извлечения данных.

В процессе работы я освоил:

использование подзапросов в блоке WHERE для фильтрации данных на основе результатов другого запроса;

применение подзапросов в блоке FROM для создания временных таблиц;

использование подзапросов в списке SELECT для вычисления дополнительных данных;

разницу между коррелированными и некоррелированными подзапросами, а также их области применения.

Полученные знания позволяют писать гибкие и сложные запросы, извлекать данные из нескольких уровней вложенности и решать задачи, требующие анализа взаимосвязей внутри базы данных. Это важный этап в изучении SQL для выполнения профессиональных аналитических задач.