

Министерство науки и высшего образования Российской Федерации

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ»
ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

197376, Санкт-Петербург, ул. проф. Попова, 5.

Факультет компьютерных технологий и информатики

Кафедра вычислительной техники

«З А Ч Т Е Н О»

_____ О.А. Жирнова

“ ” _____ 2021 г.

**ОТЧЁТ
по дисциплине «Базы данных»**

**Лабораторная работа № 5
«Работа с вложенными запросами»**

Студент группы 9308

_____ С. А.Дубенков

Санкт Петербург 2021

Цель работы: научиться писать и применять вложенные запросы [лит 1].

Используемая база данных (БД): Library.

Порядок выполнения

Упражнение 1 – использование вложенных запросов как производных таблиц

1. Выполнен запрос, использующий производные таблицы. Запрос, формирующий производную таблицу, возвращает столбец juvenile.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 включает поля adult_member_no и No_Of_Children из производного запроса и поле expr_date из таблицы adult. Выполнен запрос, использующий производную таблицу:

```
SELECT adult.member_no, children.child_num, expr_date
FROM (SELECT adult_member_no, COUNT(member_no) as
child_num
FROM juvenile
GROUP BY adult_member_no
HAVING COUNT(member_no) > 3) AS children
INNER JOIN adult on adult.member_no = chil-
dren.adult_member_no
```

Результат выполнения запроса показан на рисунке 1

100 %

Результаты

Сообщения

	member_no	child_num	expr_date
1	1	4	2006-03-19 21:32:38.513
2	3	4	2006-03-21 21:32:38.513
3	5	4	2006-03-23 21:32:38.513
4	7	4	2006-03-25 21:32:38.513
5	9	4	2006-03-27 21:32:38.513
6	11	4	2006-03-29 21:32:38.513
7	13	4	2006-03-31 21:32:38.513
8	15	4	2006-04-02 21:32:38.513
9	17	4	2006-04-04 21:32:38.513
10	19	4	2006-04-06 21:32:38.513
11	21	4	2006-04-08 21:32:38.513
12	23	4	2006-04-10 21:32:38.513
13	25	4	2006-04-12 21:32:38.513
14	27	4	2006-04-14 21:32:38.513
15	29	4	2006-04-16 21:32:38.513
16	31	4	2006-04-18 21:32:38.513
17	33	4	2006-04-20 21:32:38.513
18	35	4	2006-04-22 21:32:38.513

Результаты

Сообщения

	adult_member_no	child_num
1	1	4
2	3	4
3	5	4
4	7	4
5	9	4
6	11	4
7	13	4
8	15	4
9	17	4
10	19	4
11	21	4
12	23	4
13	25	4
14	27	4
15	29	4
16	31	4
17	33	4
18	35	4

Результаты

Сообщения

	member_no	child_num	expr_date
232	463	4	2006-06-24 21:32:38.513
233	465	4	2006-06-26 21:32:38.513
234	467	4	2006-06-28 21:32:38.513
235	469	4	2006-06-30 21:32:38.513
236	471	4	2006-07-02 21:32:38.513
237	473	4	2006-07-04 21:32:38.513
238	475	4	2006-07-06 21:32:38.513
239	477	4	2006-07-08 21:32:38.513
240	479	4	2006-07-10 21:32:38.513
241	481	4	2006-07-12 21:32:38.513
242	483	4	2006-07-14 21:32:38.513
243	485	4	2006-07-16 21:32:38.513
244	487	4	2006-07-18 21:32:38.513
245	489	4	2006-07-20 21:32:38.513
246	491	4	2006-07-22 21:32:38.513
247	493	4	2006-07-24 21:32:38.513
248	495	4	2006-07-26 21:32:38.513

Результаты

Сообщения

	adult_member_no	child_num
232	463	4
233	465	4
234	467	4
235	469	4
236	471	4
237	473	4
238	475	4
239	477	4
240	479	4
241	481	4
242	483	4
243	485	4
244	487	4
245	489	4
246	491	4
247	493	4
248	495	4

Рисунок 1

Проверка успешного формирования производной таблицы при помощи запроса

```
SELECT adult_member_no, COUNT(member_no) as child_num
FROM juvenile
GROUP BY adult_member_no
HAVING COUNT(member_no) > 3
```

2. Создание запроса производной таблице в виде двух разных запросов.

Предыдущий запрос был переписан так, чтобы он выглядел как два разных запроса, показывающие работу с производными таблицами. Написан

запрос, который возвращает поле `adult_member_no`, вычисляет количество детей для каждого взрослого читателя и возвращает те записи, в которых детей в таблице `juvenile` больше 3. Напишем еще один запрос, который возвращает значение поля `expr_date` таблицы `adult`. Перепишем запрос так, чтобы он использовал `JOIN`:

```
WITH info AS
(
  SELECT juvenile.adult_member_no, COUNT(juvenile.member_no)
  as child_num, adult.expr_date
  FROM juvenile INNER JOIN adult ON adult.member_no =
  juvenile.adult_member_no
  GROUP BY juvenile.adult_member_no, adult.expr_date
  HAVING COUNT(juvenile.member_no) > 3
)
SELECT * FROM info
```

Результат выполнения запроса показан на рисунке 2

100 %

Результаты

Сообщения

	adult_member_no	child_num	expr_date
1	1	4	2006-03-19 21:32:38.513
2	3	4	2006-03-21 21:32:38.513
3	5	4	2006-03-23 21:32:38.513
4	7	4	2006-03-25 21:32:38.513
5	9	4	2006-03-27 21:32:38.513
6	11	4	2006-03-29 21:32:38.513
7	13	4	2006-03-31 21:32:38.513
8	15	4	2006-04-02 21:32:38.513
9	17	4	2006-04-04 21:32:38.513
10	19	4	2006-04-06 21:32:38.513
11	21	4	2006-04-08 21:32:38.513
12	23	4	2006-04-10 21:32:38.513
13	25	4	2006-04-12 21:32:38.513
14	27	4	2006-04-14 21:32:38.513
15	29	4	2006-04-16 21:32:38.513
16	31	4	2006-04-18 21:32:38.513
17	33	4	2006-04-20 21:32:38.513
18	35	4	2006-04-22 21:32:38.513

100 %

Результаты

Сообщения

	adult_member_no	child_num	expr_date
232	463	4	2006-06-24 21:32:38.513
233	465	4	2006-06-26 21:32:38.513
234	467	4	2006-06-28 21:32:38.513
235	469	4	2006-06-30 21:32:38.513
236	471	4	2006-07-02 21:32:38.513
237	473	4	2006-07-04 21:32:38.513
238	475	4	2006-07-06 21:32:38.513
239	477	4	2006-07-08 21:32:38.513
240	479	4	2006-07-10 21:32:38.513
241	481	4	2006-07-12 21:32:38.513
242	483	4	2006-07-14 21:32:38.513
243	485	4	2006-07-16 21:32:38.513
244	487	4	2006-07-18 21:32:38.513
245	489	4	2006-07-20 21:32:38.513
246	491	4	2006-07-22 21:32:38.513
247	493	4	2006-07-24 21:32:38.513
248	495	4	2006-07-26 21:32:38.513

Рисунок 1

Упражнение 2 – использование подзапросов как выражений

1. Использован подзапрос с одним значением (single-value). Был написан запрос, возвращающий значения полей member.firstname, member.lastname, loanhist.isbn и loanhist.fine_paid для читателей библиотеки, заплативших максимальных штрафов за все книги.

Написан запрос, который возвращает максимальное значение поля loanhist.fine_paid

```
SELECT MAX(loanhist.fine_paid)
FROM loanhist
```

Результат выполнения запроса показан на рисунке 3

100 %	
Результаты Сообщения	
(Отсутствует имя столбца)	
1	8,00

100 %	
Результаты Сообщения	
	fine_paid
1	8,00
2	8,00
3	8,00
4	8,00
5	8,00
6	8,00
7	8,00
8	8,00
9	8,00
10	8,00
11	8,00
12	8,00
13	8,00
14	8,00
15	8,00
16	8,00
17	8,00
18	8,00

Рисунок 3

Проверка успешного возвращения максимального значения поля была выполнена при помощи запроса

```
SELECT fine_paid
FROM loanhist
ORDER BY fine_paid DESC
```

2. Использование подзапроса как части условия поиска:

2.1 Написан запрос, который соединяет таблицы member и loanhist и возвращает значения firstname, lastname, isbn и fine_paid для всех строк.

2.2 Использование запроса из шага 1 предыдущего задания как критерий выбора в предложении WHERE так, чтобы возвращались только те записи, в которых штраф

2.3 Включение ключевого слова DISTINCT.

```
SELECT DISTINCT member.firstname, member.lastname,
loanhist.isbn,
loanhist.fine_paid
FROM loanhist INNER JOIN member on member.member_no =
loanhist.member_no
```

```
WHERE fine_paid = (SELECT MAX(fine_paid) FROM
loanhist)
```

Результат выполнения запроса показан на рисунке 4

	firstname	lastname	isbn	fine_paid
1	Angela	Hightower	221	8,00
2	Clair	Rothenberg	32	8,00
3	Karl	Wolfe-Hellene	403	8,00
4	William	Erickson	30	8,00

Рисунок 4

3. Использование запросов для создания списка значений. Написан запрос к таблицам 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. Отображены те записи, которых в резерве или более 50, или менее 5. Написан запрос, возвращающий номера isbn из таблицы reservations для книг, у которых более 50 копий, или менее 5.

```
SELECT isbn, COUNT(*) as total_reserved
FROM reservation
GROUP BY isbn
HAVING COUNT(*) > 50 or COUNT(*) < 5
```

Результат выполнения запроса показан на рисунке 5

	isbn	total_reserved
1	1	197
2	43	196
3	246	196
4	288	197
5	330	196
6	533	196
7	575	197
8	617	196
9	820	196
10	862	197
11	904	196

Рисунок 5

4. Использование подзапроса с несколькими значениями. Написан внешний запрос, который возвращает поля title_no, title, isbn и Total Reserved, где Total Reserved – это число записей для каждой группы остальных полей:

4.1 Ограничено число записей, задав условие на количество копий книг, которое должно быть менее 5

4.2 Использовано ключевое слово IN, как часть предложения WHERE для списка, созданного на шаге 1 предыдущего задания

```
SELECT DISTINCT t.title_no, t.title, c.isbn, re-  
served.total_reserved  
FROM copy c  
INNER JOIN title t ON c.title_no = t.title_no  
INNER JOIN (SELECT isbn, COUNT(*) as total_reserved  
FROM reservation  
GROUP BY isbn  
HAVING COUNT(*) > 50 or COUNT(*) < 5) as reserved ON re-  
served.isbn = c.isbn  
WHERE c.isbn IN (SELECT isbn FROM reservation  
GROUP BY isbn  
HAVING COUNT(*) > 50 or COUNT(*) < 5)
```

Результат выполнения запроса показан на рисунке 6

	title_no	title	isbn	total_reserved
1	1	Last of the Mohicans	1	197
2	4	Songs of a Savoyard	533	196
3	5	Fall of the House of Usher	43	196
4	8	The Cherry Orchard	575	197
5	12	Walking	617	196
6	25	The Black Tulip	246	196
7	29	Misalliance	288	197
8	32	The Call of the Wild	820	196
9	33	The First 100,000 Prime Numbers	330	196
10	37	Treasure Island	862	197
11	41	Sense and Sensibility	904	196

Рисунок 6

Упражнение 3 – использование коррелированных подзапросов

В этом задании нужно создать запрос, использующий коррелированный подзапрос для вычисления значений, основанных на данных из внешнего запроса, и использующий это значения как часть условия сравнения.

Необходимо отобразить список читателей, имеющих сумму штрафов, превышающую 5 у. е.

1. Написан внешний запрос, возвращающий значения полей member_no и lastname. Псевдоним для таблицы member использован.

2. Написан внутренний запрос, вычисляющий общий штраф для каждого читателя. Для этого используем псевдоним для таблицы loanhist
3. Связаны поля member.member_no из внешнего запроса с полем loanhist.member_no внутреннего подзапроса.
4. Использован оператор сравнения в предикате where

```
SELECT fines.member_no, lastname, fines.total_fine
FROM member as mem
INNER JOIN (SELECT member_no, SUM(fine_assessed) as total_fine
FROM loanhist
GROUP BY member_no) as fines ON fines.member_no =
mem.member_no
WHERE fines.total_fine > 5
```

Результат выполнения запроса показан на рисунке 7

	member_no	lastname	total_fine
1	82	Chen	52,00
2	259	Hightower	234,00
3	322	LaBrie	52,00
4	502	Rudd	52,00
5	731	Brooke	26,00
6	947	Jones	182,00
7	953	Jones	78,00
8	1148	Rothenberg	208,00
9	1163	Rothenberg	78,00
10	1267	Thomas	182,00
11	1335	Young	130,00
12	1442	Chen	52,00
13	1508	Erickson	208,00
14	1588	Hightower	208,00
15	1667	LaBrie	182,00
16	1812	Perry	52,00
17	1942	Thomas	52,00
18	1988	Wolfe-Hellene	208,00

	member_no	lastname	total_fine
41	6366	Jones	156,00
42	6999	Hightower	234,00
43	7031	Jones	26,00
44	7063	Kilwine	78,00
45	7394	Wolfe-Hellene	104,00
46	7407	Wolfe-Hellene	182,00
47	7575	Erickson	130,00
48	7816	Mikovsky	156,00
49	7843	Nash	78,00
50	7952	Rudd	52,00
51	7963	Sherman	78,00
52	8226	Durkin	156,00
53	8249	Erickson	234,00
54	8412	Kilwine	52,00
55	8681	Smith	26,00
56	8924	Erickson	104,00
57	9562	Chen	52,00

Рисунок 7

Выводы: в ходе выполнения данной лабораторной работы была достигнута ее цель – были получены навыки в написании и применении вложенных запросов. Также, были получены навыки использования подзапросов как выражений, подзапросов как части условия поиска, подзапросов с несколькими значениями, коррелированных подзапросов. Для создания производной таблицы использовалась команда WITH ... AS

Список использованных источников

Методические указания к лабораторным работам / Сост.: А. В. Горячев, Н.Е. Новакова. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2008. 32 с.