

**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 № 3
«Выполнение запросов по нескольким таблицам»**

Студент группы 9308

Н.В. Яловега

Санкт Петербург 2021

Цель работы: научиться соединять данные из нескольких таблиц.

Используемая база данных (БД): Library.

Порядок выполнения

Упражнение 1 – создание списка почтовой рассылки с использованием оператора “JOIN”

Запрос для таблиц *member* и *adult*, для получения значения полей *firstname*, *middleinitial*, *lastname*, *street*, *city*, *state*, *zip*. Значения полей *firstname*, *middleinitial*, *lastname* были конкатенированы в один столбец с псевдонимом *name*:

SELECT

concat(member.firstname, '', member.middleinitial, '', member.lastname)

AS

name, adult.street, adult.city, adult.state, adult.zip

FROM

member JOIN adult ON member.member_no = adult.member_no

Результат выполнения запроса (рисунок 1)



	name	street	city	state	zip
1	Amy A Anderson	Bowery Estates	Montgomery	AL	36100
2	Brian A Anderson	Dogwood Drive	Sacramento	CA	94203
3	Daniel A Anderson	Fir Street	Washington	DC	20510-0001
4	Eva A Anderson	The Highlands	Atlanta	GA	30026
5	Gary A Anderson	James Road	Springfield	IL	62700
6	Joshua A Anderson	Larch Mont Road	Boston	MA	02138-7190
7	Katie A Anderson	New Pike Road	Trenton	NJ	08600
8	Mary Anne A Anderson	Pine Place	Albany	NY	12200-2837
9	Monica A Anderson	Race Track Road	Harrisburg	PA	17100
10	Peter A Anderson	Tamarack Road	Salt Lake City	UT	84100
11	Sarah A Anderson	Valley Road	Olympia	WA	98507
12	Stephen A Anderson	Apple Lane	Juneau	AK	99801
13	Thom A Anderson	Cannery Row	Phoenix	AZ	85011
14	Amy A Barr	Elm Street	Denver	CO	80201
15	Brian A Barr	Goshen Road	Tallahassee	FL	32301
16	Daniel A Barr	Island Drive	Honolulu	HI	96800-3017
17	Eva A Barr	King Street	Baton Rouge	LA	70800
18	Gary A Barr	Maple Street	Annapolis	MD	21400
19	Joshua A Barr	Oak Drive	Carson City	NV	89701
20	Katie A Barr	Queen Anne Drive	Salem	OR	97301

Рисунок 1

Упражнение 2 – создание списка почтовой рассылки с использованием оператора “JOIN”

Запрос по таблицам *title*, *item* и *copy*, который возвращает поля *isbn*, *copy_no*, *on_loan*, *title*, *translation* и *cover*, а также строки из таблицы *copy*, где *isbn* равен 1, 500 или 1000. Полученный набор сортируется по полю *isbn*:

SELECT

copy.isbn AS 'Isbn', *copy*.copy_no AS 'Copy', *copy*.on_loan AS 'Loan', *title*.title AS 'Title',
item.translation AS 'Trans', *item*.cover AS 'Cov'

FROM

title INNER JOIN (*copy* INNER JOIN *item* ON *item*.isbn = *copy*.isbn) ON *title*.title_no =
copy.title_no

WHERE

item.isbn=1 or *item*.isbn=500 or *item*.isbn=1000

ORDER BY isbn ASC

Результат выполнения запроса (рисунок 2)

Results		Messages				
	Isbn	Copy	Loan	Title	Trans	Cov
1	1	1	Y	Last of the Mohicans	ARABIC	HARDBACK
2	1	2	N	Last of the Mohicans	ARABIC	HARDBACK
3	1	3	N	Last of the Mohicans	ARABIC	HARDBACK
4	1	4	N	Last of the Mohicans	ARABIC	HARDBACK
5	1	5	N	Last of the Mohicans	ARABIC	HARDBACK
6	1	6	Y	Last of the Mohicans	ARABIC	HARDBACK
7	1	7	N	Last of the Mohicans	ARABIC	HARDBACK
8	1	8	N	Last of the Mohicans	ARABIC	HARDBACK
9	1	9	N	Last of the Mohicans	ARABIC	HARDBACK
10	1	10	N	Last of the Mohicans	ARABIC	HARDBACK
11	500	1	N	Frankenstein	SPANISH	HARDBACK
12	500	2	Y	Frankenstein	SPANISH	HARDBACK
13	500	3	N	Frankenstein	SPANISH	HARDBACK
14	500	4	N	Frankenstein	SPANISH	HARDBACK
15	500	5	N	Frankenstein	SPANISH	HARDBACK
16	500	6	N	Frankenstein	SPANISH	HARDBACK
17	500	7	Y	Frankenstein	SPANISH	HARDBACK
18	500	8	N	Frankenstein	SPANISH	HARDBACK
19	500	9	N	Frankenstein	SPANISH	HARDBACK
20	500	10	N	Frankenstein	SPANISH	HARDBACK

Query executed successfully. KIVYFREACT (15.0 RTM) KIVYFREACT\yalov (60) library 00:00:00 30 rows

Рисунок 2

Упражнение 3 – объединение нескольких таблиц и сортировка результатов

Запрос, возвращающий полное имя читателя *member_no* из таблицы *member*, *isbn* и *log_date* из таблицы *reservation* для читателей с номерами 250, 341 и 1675. Результат сортируется по *member_no*. Информация об этих читателях показывается вне зависимости от того, взяты ими книги или нет:

SELECT

```
member.member_no, concat(member.lastname, ' ', member.firstname, ' ',  
member.middleinitial) AS 'name', reservation.isbn, CONVERT(char(8), reservation.log_date) AS  
'date'
```

FROM

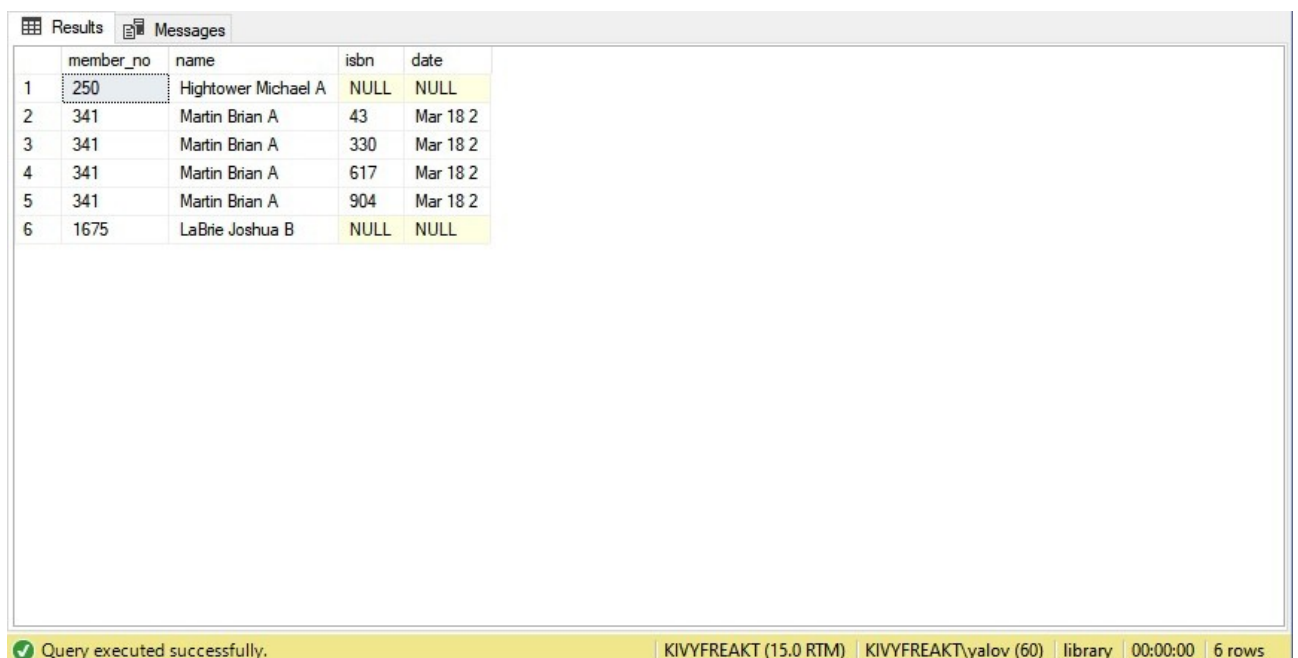
```
member LEFT OUTER JOIN reservation ON member.member_no =  
reservation.member_no
```

WHERE

```
member.member_no =250 or member.member_no = 341 or member.member_no = 1675
```

ORDER BY member_no ASC

Результат выполнения запроса (рисунок 3)



	member_no	name	isbn	date
1	250	Hightower Michael A	NULL	NULL
2	341	Martin Brian A	43	Mar 18 2
3	341	Martin Brian A	330	Mar 18 2
4	341	Martin Brian A	617	Mar 18 2
5	341	Martin Brian A	904	Mar 18 2
6	1675	LaBrie Joshua B	NULL	NULL

Query executed successfully. | KIVYFREAKT (15.0 RTM) | KIVYFREAKT\yalov (60) | library | 00:00:00 | 6 rows

Рисунок 3

Упражнение 4 – использование оператора UNION для соединения результирующих наборов

Запрос 1. Требуется определить читателей, живущих в Аризоне, у которых более двух детей посещают библиотеку:

SELECT

member.member_no, COUNT(juvenile.member_no) AS Numkids

FROM

member INNER JOIN juvenile ON member.member_no = juvenile.adult_member_no

INNER JOIN adult ON member.member_no = adult.member_no

WHERE

adult.state = 'AZ'

GROUP BY

member.member_no

HAVING COUNT(juvenile.member_no) > 2

Результат выполнения запроса (рисунок 4)



	member_no	Numkids
1	25	4
2	71	4
3	117	4
4	163	4
5	209	4
6	255	4
7	301	4
8	347	4
9	393	4
10	439	4
11	485	4
12	531	3
13	577	3
14	623	3
15	669	3
16	715	3
17	761	3
18	807	3
19	853	3
20	899	2

Query executed successfully. | KIVYFREACT (15.0 RTM) | KIVYFREACT\yalov (60) | library | 00:00:00 | 21 rows

Рисунок 4

Запрос 2. Нужно определить читателей, живущих в Калифорнии, у которых более трёх детей ходят в библиотеку:

SELECT

member.member_no, COUNT(juvenile.member_no) AS Numkids

FROM

member INNER JOIN juvenile ON member.member_no = juvenile.adult_member_no

INNER JOIN adult ON member.member_no = adult.member_no

WHERE

adult.state = 'CA'

GROUP BY

member.member_no

HAVING COUNT(juvenile.member_no) > 3

Результат выполнения запроса (рисунок 5)



The screenshot shows a database query results window with two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with two columns: 'member_no' and 'Numkids'. The table contains 11 rows of data, all with a 'Numkids' value of 4. The first row is highlighted. Below the table, a status bar indicates 'Query executed successfully.' and provides details about the database environment: 'KIVYFREACT (15.0 RTM)', 'KIVYFREACT\yalov (60)', 'library', '00:00:00', and '11 rows'.

	member_no	Numkids
1	3	4
2	49	4
3	95	4
4	141	4
5	187	4
6	233	4
7	279	4
8	325	4
9	371	4
10	417	4
11	463	4

Query executed successfully. KIVYFREACT (15.0 RTM) | KIVYFREACT\yalov (60) | library | 00:00:00 | 11 rows

Рисунок 5

Запрос 3. Объединение двух предыдущих запросов:

SELECT

member.member_no, COUNT(juvenile.member_no) AS Numkids

FROM

member INNER JOIN juvenile ON member.member_no = juvenile.adult_member_no

INNER JOIN adult ON member.member_no = adult.member_no

WHERE

adult.state = 'AZ'

GROUP BY

member.member_no

HAVING

COUNT(juvenile.member_no) > 2

UNION

SELECT

member.member_no, COUNT(juvenile.member_no) AS Numkids

FROM

member INNER JOIN juvenile ON member.member_no = juvenile.adult_member_no

INNER JOIN adult ON member.member_no = adult.member_no

WHERE

adult.state = 'CA'

GROUP BY

member.member_no

HAVING

COUNT(juvenile.member_no) > 3

Результат выполнения запроса (рисунок 6)

Results		Messages
	member_no	Numkids
1	3	4
2	25	4
3	49	4
4	71	4
5	95	4
6	117	4
7	141	4
8	163	4
9	187	4
10	209	4
11	233	4
12	255	4
13	279	4
14	301	4
15	325	4
16	347	4
17	371	4
18	393	4
19	417	4
20	439	4

Query executed successfully.
 KIVYFREACT (15.0 RTM)
 KIVYFREACT\yalov (60)
 library
 00:00:00
 32 rows

Рисунок 6

Выводы

Научились соединять данные из нескольких таблиц. Для этого более подробно изучены операторы JOIN и UNION, позволяющие объединять данные необходимым образом. Для объединения по определенному критерию данных из нескольких таблиц использовался оператор JOIN. Для объединения результирующих наборов нескольких запросов использовали оператор UNION.

Список использованных источников

1. Горячев А. В., Новакова Н. Е. Распределенные базы данных. Мет. указания к лаб. работам., СПб. Изд-во СПбГЭТУ «ЛЭТИ», 2008
2. Горячев А.В, Новакова Н.Е. Особенности разработки и администрирования приложений баз данных: учеб. пособие. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2016. 68 с.
3. Дейт К. Введение в системы баз данных. : Пер. с англ. – 6-е изд. - К.:Диалектика, 1998.