**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра САПР**

**ОТЧЕТ**

**по лабораторной работе № 3**

**по дисциплине "Базы данных"**

**Тема: "Выполнение запросов по нескольким таблицам"**

| Студенты гр. 2311 |  | Завьялов Н.С. |
| --- | --- | --- |
|  |  | Князев И.Н. |
| Преподаватель |  | Новакова Н.Е. |

Санкт-Петербург

2024

*Цель работы:* научиться соединять данные из нескольких таблиц

*Используемая база данных:* Library

**Упражнение 1 – создание списка почтовой расслки с использованием оператора “JOIN”.**  
Запрос 1. Использование JOIN:

SELECT firstname +' '+ middleinitial +' '+ lastname AS name,

street,

city,

state,

zip

FROM library.dbo.adult adu

JOIN library.dbo.member mem

ON adu.member\_no = mem.member\_no

Результаты выполнения запроса

name street city state zip

--------------------------------- --------------- --------------- ----- ----------

Amy A Anderson Bowery Estates Montgomery AL 36100

Brian A Anderson Dogwood Drive Sacramento CA 94203

Daniel A Anderson Fir Street Washington DC 20510-0001

Eva A Anderson The Highlands Atlanta GA 30026

...

Mary Anne M Sherman Race Track Road Harrisburg PA 17100

(5000 rows affected)

Completion time: 2024-09-26T22:22:12.1726388+03:00

**Упражнение 2 – объединение нескольких таблиц и сортировка результатов.**

Запрос :

SELECT copy.isbn, copy\_no, on\_loan, title, translation, cover

FROM library.dbo.title AS title

JOIN library.dbo.copy AS copy

ON title.title\_no = copy.title\_no

JOIN library.dbo.item AS item

ON item.isbn = copy.isbn

where copy.isbn = 1000 OR copy.isbn = 1500

ORDER BY copy.isbn

Результаты выполнения запроса

isbn copy\_no on\_loan title translation cover

----------- ------- ------- --------------------------------------------------------------- ----------- --------

1000 1 N Frankenstein SPANISH SOFTBACK

1000 2 Y Frankenstein SPANISH SOFTBACK

1000 3 N Frankenstein SPANISH SOFTBACK

...

1000 10 N Frankenstein SPANISH SOFTBACK

(10 rows affected)

Completion time: 2024-09-26T22:54:15.8719997+03:00

**Упражнение 3 – объединение таблиц с использованием OUTER JOIN**

Запрос 1. :

SELECT mem.firstname +' '+ mem.middleinitial +' '+ mem.lastname name,

mem.member\_no,

res.isbn,

CONVERT(char(8), res.log\_date) AS date

FROM library.dbo.member AS mem

LEFT JOIN library.dbo.reservation AS res

ON mem.member\_no = res.member\_no

WHERE mem.member\_no in (250, 341, 1675)

ORDER BY mem.member\_no

Результаты выполнения запроса

name member\_no isbn date

--------------------------------- --------- ----------- --------

Michael A Hightower 250 NULL NULL

Brian A Martin 341 43 мар 18 2

Brian A Martin 341 330 мар 18 2

Brian A Martin 341 617 мар 18 2

Brian A Martin 341 904 мар 18 2

Joshua B LaBrie 1675 NULL NULL

(6 rows affected)

Completion time: 2024-09-27T16:31:26.8132357+03:00

**Упражнение 4 – использование оператора UNION для соединения результирующих наборов.**

Запрос 1. Читатели в Аризоне, у которых более двух детей посещают библиотеку:

SELECT adu.member\_no,

COUNT(\*) AS numkids

FROM library.dbo.adult AS adu

JOIN library.dbo.juvenile AS juv

ON adu.member\_no = juv.adult\_member\_no

WHERE state = 'AZ'

GROUP BY adu.member\_no

HAVING COUNT(\*) > 2

Результаты выполнения запроса

member\_no numkids

--------- -----------

25 4

71 4

...

899 3

945 3

(21 rows affected)

Completion time: 2024-09-29T17:06:16.1937006+03:00

Запрос 2. Читатели в Калифорнии, у которых более 3х детей ходят в библиотеку:

SELECT adu.member\_no,

COUNT(\*) AS numkids

FROM library.dbo.adult AS adu

JOIN library.dbo.juvenile AS juv

ON adu.member\_no = juv.adult\_member\_no

WHERE state = 'CA'

GROUP BY adu.member\_no

HAVING COUNT(\*) > 3

Результаты выполнения запроса

member\_no numkids

--------- -----------

3 4

49 4

...

417 4

463 4

(11 rows affected)

Completion time: 2024-09-29T17:13:28.7860840+03:00

Запрос 3. Объединение запросов 1 и 2:

SELECT adu.member\_no,

COUNT(\*) AS numkids

FROM library.dbo.adult AS adu

JOIN library.dbo.juvenile AS juv

ON adu.member\_no = juv.adult\_member\_no

WHERE state = 'CA'

GROUP BY adu.member\_no

HAVING COUNT(\*) > 3

UNION

SELECT adu.member\_no,

COUNT(\*) AS numkids

FROM library.dbo.adult AS adu

JOIN library.dbo.juvenile AS juv

ON adu.member\_no = juv.adult\_member\_no

WHERE state = 'AZ'

GROUP BY adu.member\_no

HAVING COUNT(\*) > 2

Результаты выполнения запроса

member\_no numkids

--------- -----------

3 4

25 4

...

899 3

945 3

(32 rows affected)

Completion time: 2024-09-29T17:17:26.1602413+03:00

**Вывод**

В ходе выполнения лабораторной работы были получены навыки объединения таблиц и соединения результирующих наборов. Для этого были использованы следующие ключевые слова:

• <table 1> **JOIN** <table 2 > **ON** <table1.field1> <comp> <table2.field2>

где:

<table 1> и <table 2 > - объединяемые таблицы

<table1.field1> и <table2.field2> - поля таблиц, по которым проводится объединение

<comp> - оператор реляционного сравнения

Данный оператор создает таблицу, в которой присутствуют колонки как из первой так и из второй таблицы. В ней оказываются строки, у которых существует соответствие между <table1.field1> и <table2.field2>.

Также у данного оператора существуют разновидности:

INNER JOIN – в результирующую таблицу строка попадает только тогда, когда есть соответствие между <table1.field1> и <table2.field2>

LEFT OUTER JOIN - в результирующую таблицу строка попадает, если есть запись в первой (левой) таблице и не обязательно во второй (правой)

RIGHT OUTER JOIN - в результирующую таблицу строка попадает, если есть запись во второй (правой) таблице и не обязательно в первой (левой)

FULL OUTER JOIN - в результирующую таблицу строка попадает, если запись есть хотя бы в одной из таблиц

• <query1> **UNION** <query2>

где:

<query1> и<query2> - запросы

Данный оператор создаёт создает таблицу, состоящую из всех строк, полученных в первом запросе и всех строк, полученных во втором запросе.

Также у данного оператора существуют разновидности:

UNION – повторяющиеся строки исключаются

UNION ALL – повторяющиеся строки не исключаются

• Также в данной работе было использовано присваивание псевдонимов таблицам <table> **AS** <alias>

где:

<table> - название таблицы

<alias> - псевдоним

К полям таблицы можно далее обращаться по псевдониму. К примеру, если есть таблица title и мы присвоили ей псевдоним ti, то к её полю author можно обратиться следующим образом: “ti.author” .