**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»**

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

Санкт-Петербург

2024

**Цель работы:** научится соединять данные из нескольких таблиц

Упражнение 1 – создание списка почтовой рассылки с использованием оператора «JOIN». Требуется создать список рассылки читателей библиотеки. Список должен включать полное имя и информацию о месте жительства читателя.

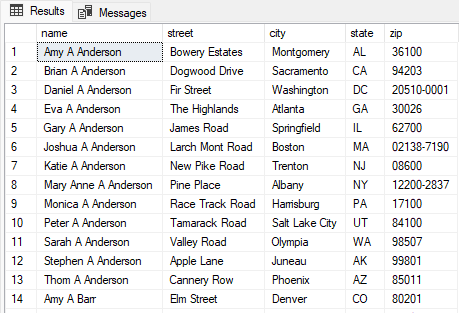
1. Задание 1.1

* Запрос

SELECT CONCAT(member.firstname,' ', member.middleinitial,' ', member.lastname) as name, adult.street, adult.city, adult.state, adult.zip

FROM member INNER JOIN adult ON member.member\_no = adult.member\_no;

* Ответ



(5000 rows)

Completion time: 2024-10-13T16:25:28.443533+03:00

Упражнение 2 – объединение нескольких таблиц и сортировка результатов. Нужно выполнить запрос по таблицам title, item и copy, который возвращал бы поля isbn, copy\_no, on\_loan, title, translation и cover, а также строки мз таблицы copy, где ISBN равен 1, 500 или 1000. Полученный набор должен быть отсортирован по полю isbn.

1. Задание 2.1

* Запрос

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

FROM title AS tl

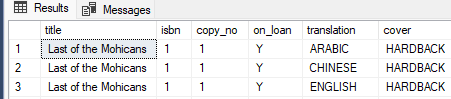
INNER JOIN copy AS cp ON tl.title\_no = cp.title\_no

INNER JOIN item AS it ON tl.title\_no=it.title\_no

WHERE cp.isbn =1 OR cp.isbn=500 OR cp.isbn=1000

ORDER BY cp.isbn ;

* Ответ



(600 rows)

Completion time: 2024-10-13T16:26:30.443533+03:00

Упражнение 3 – объединение таблиц с использованием OUTER JOIN. Необходимо выпольнить 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 отсортировать по member\_no. Показать информацию об этих читателях вне зависимости от того, взяты ими книги или нет.

1. Задание 3.1

* Запрос

SELECT CONCAT(member.firstname,' ', member.middleinitial,' ', member.lastname) as name,reservation.isbn, CONVERT(char(8), reservation.log\_date) AS log\_date

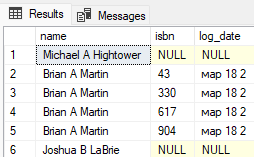
FROM member

LEFT JOIN reservation ON member.member\_no = reservation.member\_no

WHERE member.member\_no IN(250,341,1675)

ORDER BY member.member\_no;

* Запрос



(6 rows)

Completion time: 2024-10-13T17:27:32.443533+03:00

Упражнение 4 – использование оператора UNION для соединения результирующих наборов

1. Задание 4.1

* Запрос

SELECT adult.member\_no

FROM juvenile

INNER JOIN member ON juvenile.adult\_member\_no = member.member\_no

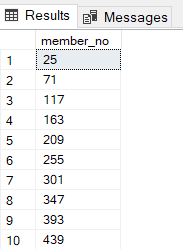
INNER JOIN adult ON member.member\_no = adult.member\_no

WHERE adult.state = 'AZ'

GROUP BY adult.member\_no

HAVING COUNT(juvenile.member\_no)>2;

* Ответ



(21 row)

Completion time: 2024-10-13T17:28:34.443533+03:00

1. Задание 4.2

* Запрос

SELECT adult.member\_no

FROM juvenile

INNER JOIN member ON juvenile.adult\_member\_no = member.member\_no

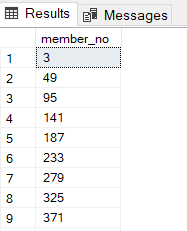
INNER JOIN adult ON member.member\_no = adult.member\_no

WHERE adult.state = 'CA'

GROUP BY adult.member\_no

HAVING COUNT(juvenile.member\_no)>2;

* Ответ



(22 rows)

Completion time: 2024-10-13T18:29:36.443533+03:00

1. Задание 4.3

* Запрос

SELECT adult.member\_no

FROM juvenile

INNER JOIN member ON juvenile.adult\_member\_no = member.member\_no

INNER JOIN adult ON member.member\_no = adult.member\_no

WHERE adult.state = 'AZ'

GROUP BY adult.member\_no HAVING COUNT(juvenile.member\_no)>2

UNION

SELECT adult.member\_no

FROM juvenile INNER JOIN member ON juvenile.adult\_member\_no = member.member\_no

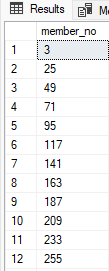
INNER JOIN adult ON member.member\_no = adult.member\_no

WHERE adult.state = 'CA'

GROUP BY adult.member\_no

HAVING COUNT(juvenile.member\_no)>2;

* Ответ



(43 rows)

Completion time: 2024-10-13T18:30:38.443533+03:00

**Вывод**:  
В ходе выполнения лабораторной работы я научился объединять данные из нескольких таблиц с использованием различных типов соединений (JOIN) в языке SQL.

В процессе работы я освоил:

* использование команды INNER JOIN для выборки записей, имеющих совпадения в обеих таблицах;
* применение LEFT JOIN и RIGHT JOIN для выборки данных из одной таблицы с учетом возможного отсутствия соответствий в другой таблице;
* использование FULL OUTER JOIN для объединения всех записей из обеих таблиц;
* работу с условием соединения для выбора данных по определенным критериям.

Полученные знания позволяют создавать запросы, которые эффективно извлекают взаимосвязанную информацию из нескольких таблиц, обеспечивая целостное представление данных. Это является важным шагом в изучении основ реляционных баз данных и подготовки к более сложным операциям анализа данных.