**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра систем автоматизированного проектирования (САПР)**

**отчет**

**по лабораторной работе № 4**

**по дисциплине «Базы данных»**

Тема: **«Модификация данных»**

|  |  |  |
| --- | --- | --- |
| Студенты гр. 3311 | Аршин А. Д  Баймухамедов Р. Р.  Пасечный Л. В. |  |
| Преподаватель | Новакова Н. Е. |  |

Санкт-Петербург

2025

**Цель работы**

Научиться выполнять команды вставки, удаления и обновления данных. В лабораторной работе используется БД Library

**Упражнение 1**

Применение команды INSERT. В этом задании необходимо использовать команду INSERT для добавления строк в таблицы БД Library. После выполнения вставки данных необходимо написать запрос для проверки сделанных изменений.

Вставим строку в таблицу item, которая представляет книги из собрания библиотеки

*Запрос*

INSERT INTO item (isbn, title\_no, cover, loanable, translation)

VALUES(10001,8,'HARDBACK','Y','ENGLISH'),(10101,8,'SOFTBACK','Y','ENGLISH')

SELECT isbn,title\_no,cover,loanable,translation

FROM item

ORDER BY isbn DESC

*Результат*

isbn title\_no cover loanable translation

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

10101 8 SOFTBACK Y ENGLISH

10001 8 HARDBACK Y ENGLISH

1000 50 SOFTBACK Y SPANISH

999 50 SOFTBACK Y PORTUGES

998 50 SOFTBACK Y LATIN

…

(Затронуто строк: 1002)

Время выполнения: 2025-09-24T22:53:49.8941173+03:00

Далее вставим строку в таблицу copy, которая содержит данные о копиях книг в собрании библиотек

*Запрос*

INSERT INTO copy(isbn, copy\_no, title\_no, on\_loan)

VALUES(10001, 1, 8, 'N')

SELECT isbn, copy\_no, title\_no, on\_loan

FROM copy

WHERE isbn = 10001

*Результат*

isbn copy\_no title\_no on\_loan

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

10001 1 8 N

(Затронута 1 строка)

Время выполнения: 2025-09-24T23:00:19.7504522+03:00

Определим язык, на который был переведен экземпляр книги из собрания библиотек

*Запрос*

SELECT translation

FROM item

WHERE isbn = 10001 AND title\_no = 8

*Результат*

translation

-----------

ENGLISH

(Затронута 1 строка)

Время выполнения: 2025-09-24T23:02:51.4271808+03:00

**Упражнение 2**

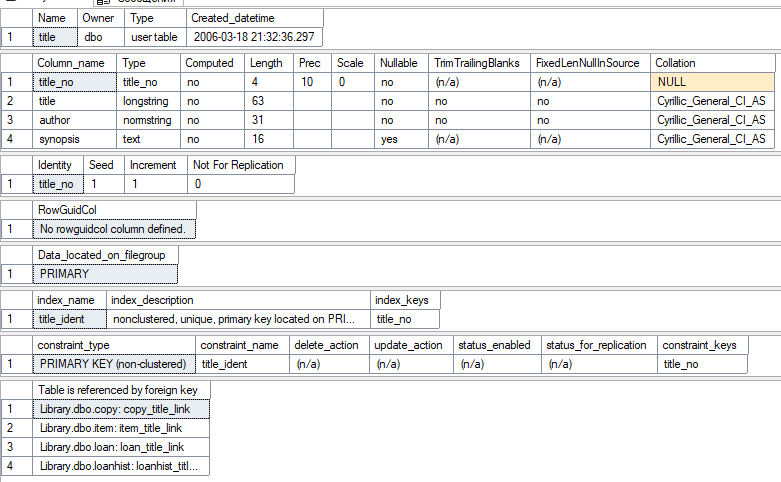
Использование команды INSERT с ключевым словом DEFAULT

Определим столбцы, для которых разрешено значение null. Выполним системную хранимую процедуру sp\_help для того, чтобы определить какие столбцы в таблице title допускает значение null. Для этих столбцов, а также для столбцов со значением DEFAULT не обязательно указывать значение при добавлении строк. Значения полей со свойством IDENTITY формируются автоматически

*Запрос*

exec sp\_help title

*Результат*



Время выполнения: 2025-09-24T23:10:45.9364798+03:00

Вставим значения в таблицу title. Вставьте строку в таблицу title для книги The Art of Lawn Tennis автора William T. Tilden. Используйте ключевое слово DEFAULT для тех полей, которые разрешают значение null или имеют значение DEFAULT. Не указывайте значение для поля title\_no поскольку это поле имеет свойство IDENTITY. Напишем и выполним запрос, подтверждающий факт вставки

*Запрос*

INSERT INTO title

VALUES('The Art of Lawn Tennis', 'William T. Tilden', DEFAULT)

SELECT title\_no, title, author, synopsis

FROM title

WHERE title = 'The Art of Lawn Tennis'

*Результат*

title\_no title author synopsis

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

51 The Art of Lawn Tennis William T. Tilden NULL

(Затронуто строк: 1)

Время выполнения: 2025-09-24T23:29:54.5186797+03:00

Определим последнее использование значения IDENTITY. Напишем запрос для определения значения поля title\_no для заголовка, добавленного в предыдущем задании

*Запрос*

SELECT title\_no

FROM title

WHERE title = 'The Art of Lawn Tennis' AND author = 'William T. Tilden'

*Результат*

title\_no

-----------

51

(Затронуто строк: 1)

Время выполнения: 2025-09-24T23:31:08.5881707+03:00

Получим последнюю вставленную запись в таблице title. Напишем запрос , возвращающий последнюю запись, вставленную в таблицу title. Используем результат предыдущего запроса.

*Запрос*

SELECT title\_no, title, author, synopsis

FROM title

WHERE title\_no = 51

*Результат*

title\_no title author synopsis

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

51 The Art of Lawn Tennis William T. Tilden NULL

(Затронута 1 строка)

Время выполнения: 2025-09-24T23:35:00.4129267+03:00

Добавим новых записей в таблицу title. Вставим в таблицу title записи для названия книги Riders of the Purple Saga автора Zane Grey. Укажем список полей и соответствующие им значения для полей, не разрешающих null и не имеющих значений DEFAULT. Проверим вставки значений в таблицу title. Напишем и выполним запрос для проверки того, что новое название книги и автор были успешно добавлены.

*Запрос*

INSERT INTO title

VALUES('Rider of the Purple Saga', 'Zane Grey', DEFAULT)

SELECT TOP 3 title\_no, title, author, synopsis

FROM TITLE

ORDER BY title\_no DESC

*Результат*

title\_no title author synopsis

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

52 Rider of the Purple Sage Zane Grey NULL

51 The Art of Lawn Tennis William T. Tilden NULL

50 Frankenstein Mary Wollstonecraft Shelley NULL

(Затронуто строк: 3)

Время выполнения: 2025-09-22T15:37:42.8304673+03:00

**Упражнение 3**

Использование команды INSERT с ключевыми словами DEFAULT VALUES

Создадим новую таблицу sample1. Выполним запрос для создания новой таблицы sample1 в базе данных library.

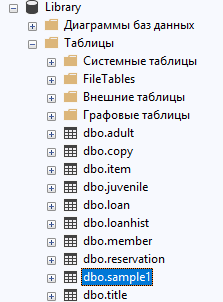
*Запрос*

CREATE TABLE.sample1(

Cust\_id int NOT NULL IDENTITY(100,5)

,Name char(10) NULL)

*Результат*



Выполнение команд успешно завершено.

Время выполнения: 2025-09-24T23:48:17.4891391+03:00

Вставим запись со значениями по умолчанию в таблицу sample1

*Запрос*

INSERT INTO sample1

DEFAULT VALUES

SELECT Cust\_id, Name

FROM sample1

*Результат*

Cust\_id Name

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

100 NULL

(Затронута 1 строка)

Время выполнения: 2025-09-24T23:54:35.8884982+03:00

**Упражнение 4**

Использование команды DELETE

Напишем запрос, возвращающий запись из таблицы item. Свойства записи: мягкий переплет, isbn 10101, название – “The Cherry Orchard”, title\_no 8.

Запрос

SELECT isbn, title\_no, translation, cover, loanable

FROM item

WHERE cover = 'SOFTBACK' AND isbn = 10101 AND title\_no = 8

Результат

isbn title\_no translation cover loanable

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

10101 8 ENGLISH SOFTBACK Y

(Затронута 1 строка)

Время выполнения: 2025-09-25T00:01:36.2915687+03:00

Удалим указанную запись

*Запрос*

DELETE FROM item

WHERE cover = 'SOFTBACK' AND isbn = 10101 AND title\_no = 8

*Результат*

(Затронута 1 строка)

Время выполнения: 2025-09-25T00:03:53.5934040+03:00

**Упражнение 5**

Использование команды UPDATE

Получим записи, которую следует обновить. Напишем запрос, возвращающий пользователя библиотеки с номером 507 из таблицы member.

*Запрос*

SELECT member\_no, lastname, firstname, middleinitial, photograph

FROM member

WHERE member\_no = 507

*Результат*

member\_no lastname firstname middleinitial photograph

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

507 Rudd Katie A NULL

(Затронута 1 строка)

Время выполнения: 2025-09-25T00:06:52.0024777+03:00

Обновим запись. Перезапишем запрос так, чтобы она изменяла фамилию указанного читателя

*Запрос*

UPDATE member

SET lastname = 'Raf\_Sasha\_Lenya'

WHERE member\_no = 507

SELECT member\_no, lastname, firstname, middleinitial, photograph

FROM member

WHERE member\_no = 507

*Результат*

(Затронута 1 строка)

member\_no lastname firstname middleinitial photograph

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

507 Raf\_Sasha\_Lenya Katie A NULL

(Затронута 1 строка)

Время выполнения: 2025-09-25T00:09:49.6831105+03:00

**Упражнение 6**

Изменение данных на основе информации других таблиц

Добавим нового юного читателя в базу данных

*Запрос*

BEGIN TRANSACTION

SET IDENTITY\_INSERT member ON

INSERT member(member\_no, lastname, firstname, middleinitial)

VALUES (16101, 'Walters', 'B.', 'L')

SET IDENTITY\_INSERT member OFF

INSERT juvenile

VALUES (16101, 1, DATEADD(YY,-18,DATEADD(DD,-1,GETDATE())))

COMMIT TRANSACTION

*Результат*

(Затронута 1 строка)

Время выполнения: 2025-09-25T00:21:16.4374940+03:00

Получим записи, которые должны быть перенесены из таблицы juvenile в таблицу adult. Напишем запрос, возвращающий значения полей member\_no из таблицы juvenile и street, city, state, zip и phone\_no из таблицы adult. Также включите в запрос текущую дату плюс 1 год. Соответствующее выражение: DATEADD(YY,1,GETDATE()). Свяжем талицы по полю juveline.adult\_member\_no = adult.member\_no. Добавим выражение WHERE, чтобы показать только тех читателей из таблицы juvenile, котороым более 18 лет. Используем функцию DATEADD

*Запрос*

SELECT jv.member\_no, ad.street, ad.city, ad.state, ad.zip, ad.phone\_no

FROM juvenile jv

INNER JOIN adult ad

ON jv.adult\_member\_no = ad.member\_no

WHERE DATEADD(YEAR,18,jv.birth\_date) < GETDATE()

*Результат*

member\_no street city state zip phone\_no

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

2 Bowery Estates Montgomery AL 36100 NULL

7126 Bowery Estates Montgomery AL 36100 NULL

8524 Bowery Estates Montgomery AL 36100 NULL

9506 Bowery Estates Montgomery AL 36100 NULL

16101 Bowery Estates Montgomery AL 36100 NULL

9508 Dogwood Drive Sacramento CA 94203 NULL

8526 Dogwood Drive Sacramento CA 94203 NULL

7128 Dogwood Drive Sacramento CA 94203 NULL

4 Dogwood Drive Sacramento CA 94203 NULL

…

(Затронуто строк: 5001)

Время выполнения: 2025-09-25T00:48:33.0289490+03:00

Вставим записи в таблицу juvenile из таблицы adult

*Запрос*

INSERT INTO adult (member\_no, street, city, state, zip, phone\_no, expr\_date)

SELECT jv.member\_no, ad.street, ad.city, ad.state, ad.zip, ad.phone\_no, jv.birth\_date

FROM juvenile AS jv

INNER JOIN adult AS ad

ON jv.adult\_member\_no = ad.member\_no

WHERE DATEADD(YEAR,18,jv.birth\_date) < GETDATE()

*Результат*

(Затронуто строк: 5001)

Время выполнения: 2025-09-25T01:03:00.9916474+03:00

Напишем запрос, подтверждающий добавление читателя с номером 16101 в таблицу adult

*Запрос*

SELECT member\_no, expr\_date

FROM adult

WHERE member\_no = 16101

*Результат*

member\_no expr\_date

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

16101 2007-09-24 00:21:16.423

(Затронута 1 строка)

Время выполнения: 2025-09-25T01:17:12.6435016+03:00

Определим подлежащих удалению записей из таблицы juvenile. Напишем запрос, который связывает таблицы juvenile и adult следующим запрос juvenile.member\_no = adult.member\_no и удалим полученные записи

*Запрос*

DELETE FROM juvenile

FROM juvenile

JOIN adult

ON juvenile.member\_no = adult.member\_no

*Результат*

(Затронуто строк: 5001)

Время выполнения: 2025-09-25T01:30:11.7184528+03:00

Убедимся, что читатель 16101 был удален из таблицы juvenile

*Запрос*

SELECT member\_no, adult\_member\_no, birth\_date

FROM juvenile

WHERE member\_no = 16101

*Результат*

member\_no adult\_member\_no birth\_date

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

(Затронуто строк: 0)

Время выполнения: 2025-09-25T01:31:38.5152616+03:00

**Вывод**

В ходе выполнения лабораторной работы были успешно освоены основные операции манипулирования данными в SQL на примере базы данных "Library". Были получены практические навыки работы с командами вставки (INSERT), обновления (UPDATE) и удаления (DELETE) данных.

Освоена техника добавления новых записей в таблицы различными способами: как одиночными операциями с явным указанием значений, так и массовой вставкой данных с помощью оператора SELECT. Научились использовать значения по умолчанию (DEFAULT) для необязательных полей и работать с автоинкрементными полями (IDENTITY).

Приобретены навыки безопасного изменения и удаления данных с использованием условий WHERE для точного отбора записей. Особое внимание уделено работе с транзакциями, что позволяет гарантировать целостность данных при выполнении сложных многотабличных операций.

Важным результатом стало освоение техники переноса данных между связанными таблицами с использованием соединений (JOIN), что является частой практической задачей в реальных базах данных. На примере переноса подростков в таблицу взрослых читателей отработан комплексный подход к управлению данными.

Полученные навыки позволяют эффективно работать с реляционными базами данных и являются фундаментом для дальнейшего изучения более сложных аспектов SQL и проектирования информационных систем.