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

Кафедра Систем автоматизированного проектирования

ОТЧЕТ

по лабораторной работе №4 по дисциплине «Базы данных»

Тема: Модификация данных

Студенты гр. 2308	 Рымарь М.И.
	 Мелихов М.А.
	 Придчин В.Е.
Преподаватель	Горяинов С.В.

Санкт-Петербург

Цель работы

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

Выполнение работы

Упражнение 1 — применение команды INSERT. В задании необходимо использовать команду INSERT для добавления строк в таблицы БД Library. После выполнения вставки данных необходимо написать запрос для проверки сделанных изменений

Запрос 1: вставка строк в таблицу item. Проверка запросом.

Результат выполнения запроса показан на рисунке 1.

	isbn	title_no	translation	cover	loanable
	10001	8	ENGLISH	HARDBACK	Y
	10101	8	ENGLISH	SOFTBACK	Y
(3	(затронуто строк:		2)		
(3	атронут	строк:	2)		

Рисунок 1 – Вставка значений в таблицу item

Время выполнения: 2024-10-21T15:08:19.9334619+03:00

Запрос 2: вставка строк в таблицу сору, которая содержит данные о копиях книг в собрании библиотеки. Проверка запросом.

```
INSERT INTO dbo.copy
(isbn, copy_no, title_no, on_loan)
VALUES (10001, 1, 8, 'N');
SELECT * FROM copy
```

```
WHERE isbn = 10001;
```

Результат выполнения запроса показан на рисунке 2.

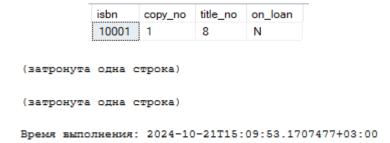


Рисунок 2 – Вставка значений в таблицу сору

Запрос 3: определить язык, на который был переведён экземпляр книги из собрания библиотеки.

```
SELECT translation FROM item
WHERE isbn = 10001;
```

Результат выполнения запроса показан на рисунке 3.

```
translation ENGLISH

(затронута одна строка)

Время выполнения: 2024-10-21Т15:10:38.9600117+03:00

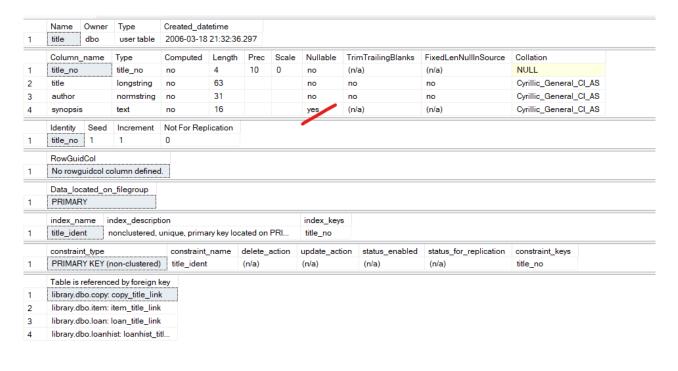
Рисунок 3 — Результат выполнения запроса
```

Упражнение 2 — использование INSERT с ключевым словом DEFAULT

Запрос 1: определить столбцы, для которых разрешено значение NULL.

```
EXEC sp_help 'dbo.title'
```

Результат выполнения запроса представлен на рисунке 4.



Время выполнения: 2024-10-21T15:22:51.3856780+03:00

Рисунок 4 – Результат выполнения запроса sp_help

Запрос 2: вставка значений в таблицу title.

```
INSERT INTO title (title, author, synopsis)
VALUES ('The Art of Lawn Tennis', 'William T. Tilden',
DEFAULT);
SELECT * FROM title
ORDER BY title_no DESC;
```

Результат выполнения запроса представлен на рисунке 5. Не трогаем поле title_no, так как оно нумеруется автоматически (обладает свойством IDENTITY).

	title_no	title	author	synopsis
1	51	The Art of Lawn Tennis	William T. Tilden	NULL
2	50	Frankenstein	Mary Wollstonecraft Shelley	NULL
3	49	Julius Caesar's Commentaries on the Gallic War	Julius Caesar	NULL
4	48	History of the Decline and Fall of the Roman Em	Edward Gibbon	NULL
5	47	The Crossing	Winston Churchill	NULL
^	46	The Committee Mandage of MCIII and Obeliance	MEIII Ch-I	NII II I

(затронута одна строка) (затронуто строк: 51)

Время выполнения: 2024-10-21T15:23:48.8537331+03:00

Рисунок 5 – Результат выполнения запроса

Запрос 3: определить последнее использованное значение поля title_no.

```
SELECT TOP(1) title_no FROM title
ORDER BY title no DESC;
```

Результат выполнения запроса представлен на рисунке 6.



(затронута одна строка)

Время выполнения: 2024-10-21T15:26:33.6198270+03:00

Рисунок 6 – Результат выполнения запроса

Запрос 4: получить последнюю вставленную запись в таблицу title.

```
SELECT TOP(1) * FROM title ORDER BY title no DESC;
```

Результат выполнения запроса представлен на рисунке 7.



(затронута одна строка)

Время выполнения: 2024-10-21T15:26:33.6198270+03:00

Рисунок 7 – Результат выполнения запроса

Запрос 5: добавить новую запись в таблицу title.

```
INSERT INTO title (title, author)
VALUES ('Riders of the Purple Sage', 'Zane Grey');
SELECT TOP(1) * FROM title
ORDER BY title no DESC
```

Результат выполнения запроса представлен на рисунке 8.

		title_no	title	author	synopsis
1	1	52	Riders of the Purple Sage	Zane Grey	NULL

```
(затронута одна строка)
(затронута одна строка)
Время выполнения: 2024-10-21T15:28:19.3640813+03:00
```

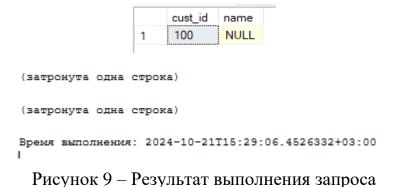
Рисунок 8 – Результат выполнения запроса

Упражнение 3 — использование команды INSERT с ключевыми словами DEFAULT VALUES

Запрос: создание новой таблицы с дефолтными значениями.

```
USE LIBRARY
CREATE TABLE sample1 (
      cust_id int NOT NULL IDENTITY (100, 5),
      name char(10) NULL);
INSERT INTO sample1 DEFAULT VALUES;
SELECT * FROM sample1
```

Результат выполнения запроса представлен на рисунке 9.



They new y Tesymbra Bentembresher sampeet

Упражнение 4 – использование команды **DELETE**

Запрос: удалить вставленную ранее запись из таблицы item со следующими характеристиками: мягкий переплёт, isbn 10101, название — «The Cherry Orchard», title_no 8.

```
DELETE FROM item
WHERE isbn = 10101
```

Результат выполнения запроса представлен на рисунке 10.

```
(затронута одна строка)

Время выполнения: 2024-09-17T23:33:17.3575075+03:00
```

Рисунок 10 – Результат выполнения удаления записи

Упражнение 5 – использование команды UPDATE

Запрос 1: получение записи, которую следует обновить — пользователь библиотеки с номером 507.

```
SELECT * FROM member
WHERE member_no = 507
```

Результат выполнения запроса показан на рисунке 11.

	member_no	lastname	firstname	middleinitial	photograph
1	507	Rudd	Katie	Α	NULL

Рисунок 11 – Результат выполнения запроса

Запрос 2: обновление фамилии пользователя библиотеки с номером 507.

```
UPDATE member

SET lastname = 'White'

WHERE member_no = 507;

SELECT * FROM member

WHERE member_no = 507
```

Результат выполнения запроса показан на рисунке 12.

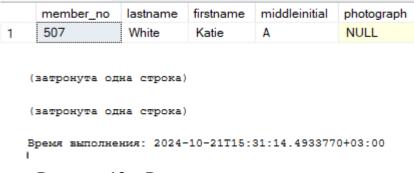


Рисунок 12 – Результат выполнения запроса

Упражнение 6 – изменение данных на основе информации других таблиц

Запрос 1: добавление нового юного читателя в базу данных.

```
USE library;
BEGIN TRANSACTION;
SET IDENTITY_INSERT member ON;
INSERT INTO member (member_no, lastname, firstname, middleinitial)
VALUES (16101, 'Walters', 'B.', 'L');
SET IDENTITY_INSERT member OFF;
INSERT INTO juvenile (member_no, adult_member_no, birth_date)
VALUES (16101, 1, DATEADD(YY, -18, DATEADD(DD, -1, GETDATE())));
COMMIT TRANSACTION;
```

Результат выполнения запроса представлен на рисунке 13. Построчное объяснение запроса:

Строка 1: USE library;: Переключается на базу данных library.

Строка 2: BEGIN TRANSACTION;: Начинает транзакцию. Транзакция - это набор операций, которые выполняются как единое целое. Если какая-либо операция в транзакции завершается неудачей, вся транзакция отменяется.

Строка 3: SET IDENTITY_INSERT member ON;: Включает возможность вставки пользовательских значений в поле со свойством IDENTITY. Обычно значения IDENTITY генерируются автоматически, но эта команда позволяет нам вставить собственное значение для member_no.

Строка 4: INSERT INTO member (member_no, lastname, firstname, middleinitial) VALUES (16101, 'Walters', 'B.', 'L');: Вставляет новую запись в таблицу member с указанными значениями для member_no, lastname, firstname и middleinitial.

Строка 5: SET IDENTITY_INSERT member OFF;: Отключает возможность вставки пользовательских значений в поле со свойством IDENTITY.

Строка 6: INSERT INTO juvenile (member_no, adult_member_no, birth_date) VALUES (16101, 1, DATEADD(YY, -18, DATEADD(DD, -1, GETDATE())));: Вставляет новую запись в таблицу juvenile с указанными значениями для member_no, adult_member_no и birth_date.

Строка 7: COMMIT TRANSACTION;: Завершает транзакцию и сохраняет изменения в базе данных.

Этот запрос выполняет следующие действия: добавляет нового члена с номером 16101, фамилией Walters, именем В. и отчеством L. в таблицу member. Добавляет новую запись о юном читателе с номером 16101, соответствующим номером взрослого члена 1 и датой рождения, которая на 18 лет меньше текущей даты, в таблицу juvenile.

```
(затронута одна строка)
(затронута одна строка)
Время выполнения: 2024-09-17T23:53:21.0352218+03:00
```

Рисунок 13 – Результат добавления читателя

Запрос 2: получение записей, которые должны быть перенесены из таблицы juvenile в таблицу adult.

```
SELECT j.member_no, a.street, a.city, a.state, a.zip,
a.phone_no, DATEADD(YY, 1, GETDATE()) AS next_year
FROM juvenile AS j

JOIN adult AS a ON j.adult_member_no = a.member_no
WHERE DATEADD(YY, -18, j.birth date) < GETDATE()</pre>
```

Результат выполнения запроса показан на рисунке 14. Фильтрация установлена по читателям, которым больше 18 лет.

	member no	street	city	state	zip	phone no	next year
1	2	Bowery Estates	Montgomery	AL	36100	NULL	2025-09-18 00:00:53.060
2	7126	Bowery Estates	Montgomery	AL	36100	NULL	2025-09-18 00:00:53.060
3	8524	Bowery Estates	Montgomery	AL	36100	NULL	2025-09-18 00:00:53.060
4	9506	Bowery Estates	Montgomery	AL	36100	NULL	2025-09-18 00:00:53.060
5	16101	Bowery Estates	Montgomery	AL	36100	NULL	2025-09-18 00:00:53.060
6	9508	Dogwood Drive	Sacramento	CA	94203	NULL	2025-09-18 00:00:53.060
7	8526	Dogwood Drive	Sacramento	CA	94203	NULL	2025-09-18 00:00:53.060
8	7128	Dogwood Drive	Sacramento	CA	94203	NULL	2025-09-18 00:00:53.060
9	4	Dogwood Drive	Sacramento	CA	94203	NULL	2025-09-18 00:00:53.060
10	6	Fir Street	Washington	DC	20510-0001	NULL	2025-09-18 00:00:53.060
11	7130	Fir Street	Washington	DC	20510-0001	NULL	2025-09-18 00:00:53.060
12	8528	Fir Street	Washington	DC	20510-0001	NULL	2025-09-18 00:00:53.060
13	9510	Fir Street	Washington	DC	20510-0001	NULL	2025-09-18 00:00:53.060
14	9512	The Highlands	Atlanta	GA	30026	NULL	2025-09-18 00:00:53.060
15	8530	The Highlands	Atlanta	GA	30026	NULL	2025-09-18 00:00:53.060

(затронуто строк: 5000)

Время выполнения: 2024-10-21T15:32:48.6642239+03:00

Рисунок 14 – Результат выполнения запроса

Запрос 3: вставка записей в таблицу juvenile из таблицы adult.

```
INSERT INTO adult (member_no, street, city, state, zip,
phone_no, expr_date)

SELECT j.member_no, a.street, a.city, a.state, a.zip,
a.phone_no, GETDATE()

FROM juvenile AS j

JOIN adult AS a ON j.adult_member_no = a.member_no

WHERE DATEADD(YY, -18, j.birth_date) < GETDATE();

SELECT * FROM adult WHERE member no = 16101;</pre>
```

Результат выполнения запроса показан на рисунке 15.

00,7,0,10,10,	⊒в Сооощения					
member_no	street	city	state	zip	phone_no	expr_date
16101	Bowery Estates	Montgomery	AL	36100	NULL	2024-09-18 00:07:45.460

(затронуто строк: 5000)

Время выполнения: 2024-10-21T15:32:48.6642239+03:00

Рисунок 15 – Результат выполнения запроса

Запрос 4: Определение подлежащих удалению записей из таблицы juvenile.

```
SELECT j.member_no
FROM juvenile AS j

JOIN adult AS a ON j.member_no = a.member_no
```

Результат выполнения запроса показан на рисунке 16.

	member_no
1	2
2	4
3	6
4	8
5	10
6	12
7	14
8	16
9	18
10	20
11	22

(затронуто строк: 5000)

Время выполнения: 2024-10-21Т15:32:48.6642239+03:00

Рисунок 16 – Результат выполнения запроса

Запрос 5: Удаление записей из таблицы juvenile.

```
DELETE FROM juvenile
WHERE member_no IN (
SELECT j.member_no FROM juvenile AS j
JOIN adult AS a ON j.member_no = a.member_no
);
SELECT * FROM juvenile WHERE member_no = 16101
```

Результат выполнения запроса показан на рисунке 17.

```
(затронуто строк: 5000)
(затронуто строк: 0)
Время выполнения: 2024-10-21T15:40:43.9546945+03:00
```

member_no adult_member_no birth_date

Рисунок 17 – Результат выполнения запроса

Выводы

В ходе выполнения лабораторной работы были успешно применены команды INSERT, DELETE и UPDATE для добавления, удаления и обновления данных в базе данных Library. Запросы были составлены правильно, что позволило выполнить поставленные задачи. Особое внимание уделялось использованию ключевых слов DEFAULT, DEFAULT VALUES и IDENTITY для автоматической генерации значений и установки значений по умолчанию. Выполнение работы способствовало закреплению полученных теоретических знаний и развитию практических навыков работы с базами данных, в частности с использованием языка SQL и с управлением данными на разных уровнях.

Кроме того, работа позволила глубже понять структуру и взаимосвязи таблиц в базе данных, а также продемонстрировала важность правильного проектирования и нормализации данных для обеспечения целостности и эффективности базы данных. Полученные знания и навыки станут основой для дальнейшего изучения и применения технологий управления данными в различных практических задачах.