**Лабораторная работа № 5**

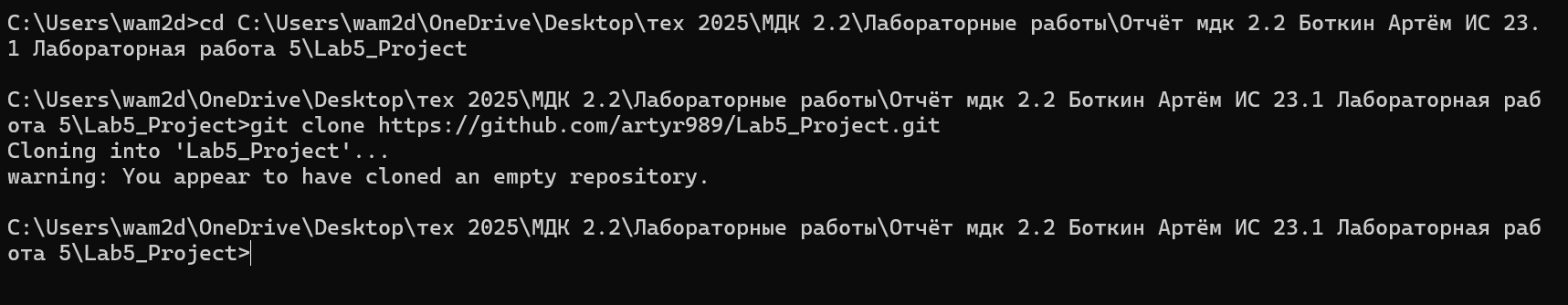
**Тема:** Разработка и интеграция модулей проекта (командная работа)

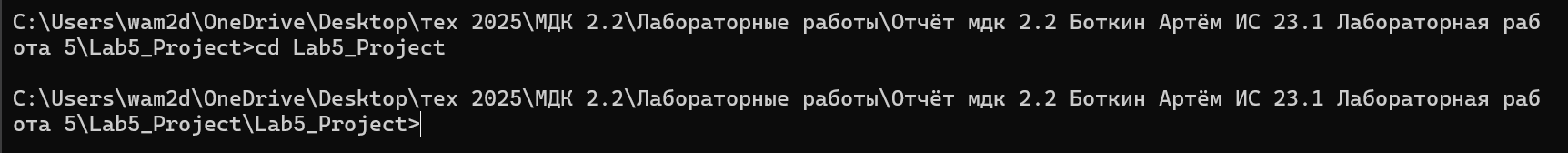
**Цель работы**

Научиться командной разработке программного проекта с использованием MS Access и Git.  
Студенты осваивают:

* создание модулей базы данных;
* разработку SQL-запросов для работы с таблицами;
* интеграцию модулей в единый проект;
* ведение проекта в системе контроля версий (Git) с настройкой .gitignore.

**Клонирование репозитория на компьютер**

****

****

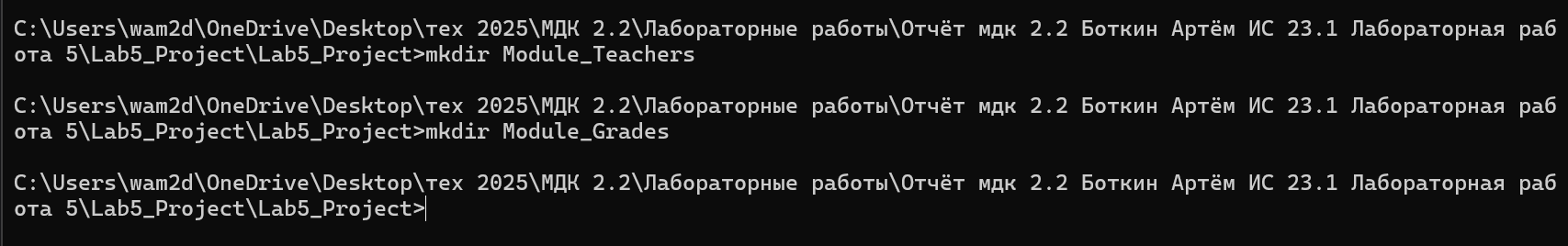
**СОЗДАНИЕ СТРУКТУРЫ ПРОЕКТА**

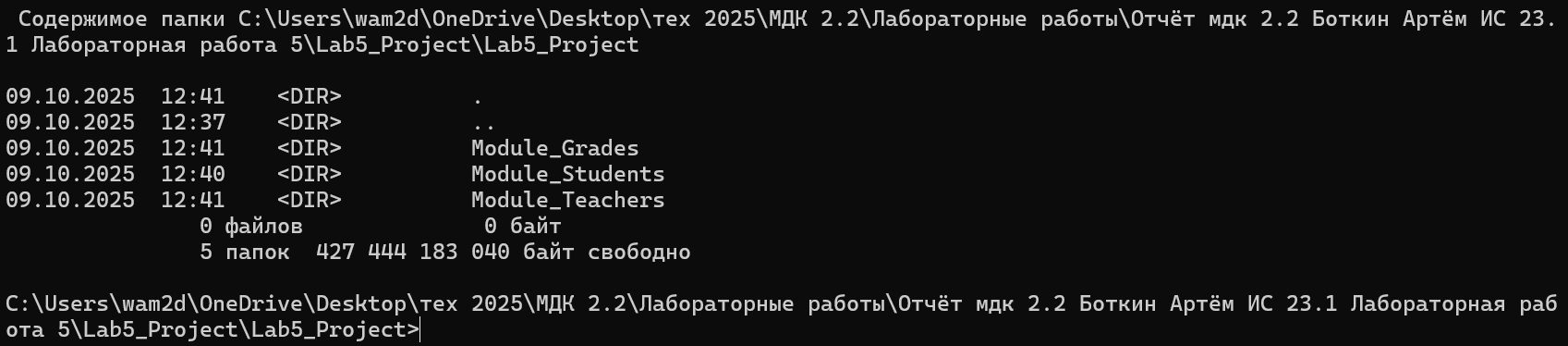
Создание папок для модулей

mkdir Module\_Students

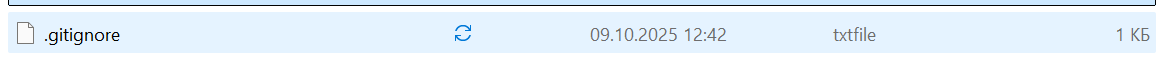
mkdir Module\_Teachers

mkdir Module\_Grades





Создание файла .gitignore

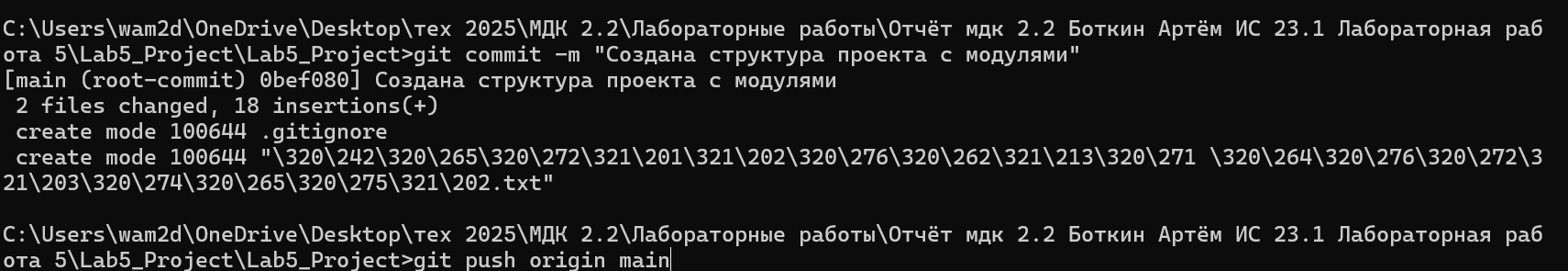


Первоначальная настройка Git

git add .

git commit -m "Создана структура проекта с модулями"

git push origin main

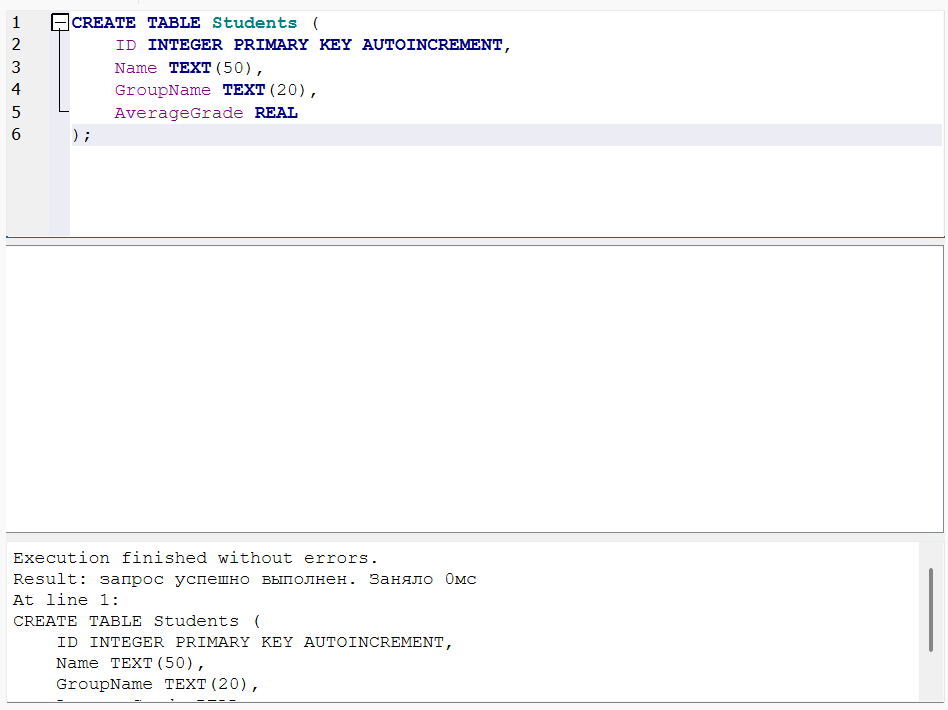


**РАЗРАБОТКА МОДУЛЕЙ В SQLITE**

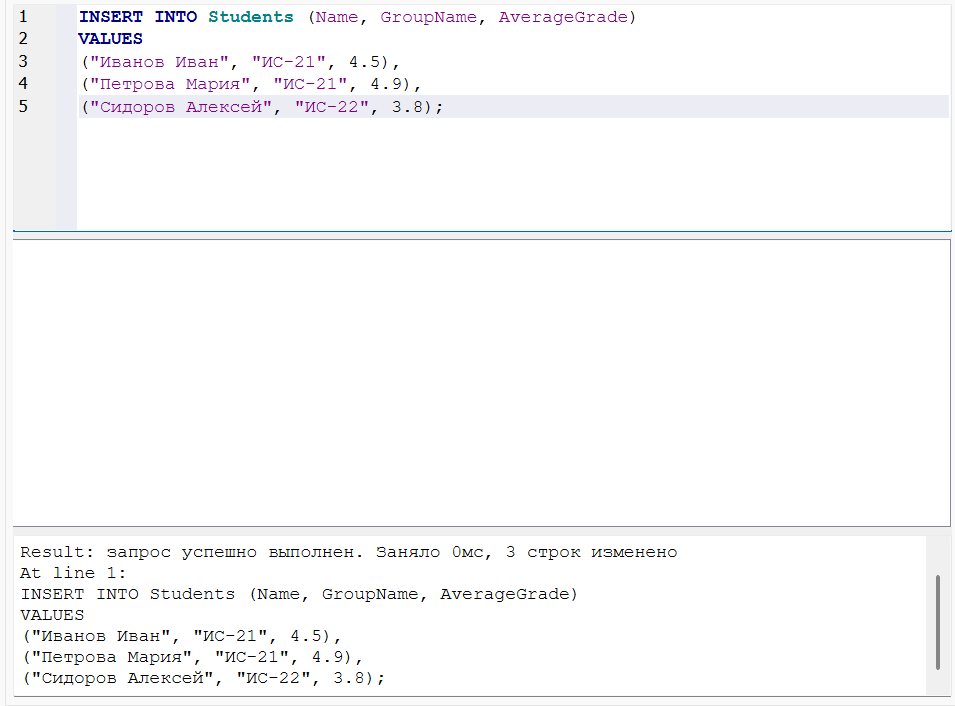
МОДУЛЬ 1: STUDENTS

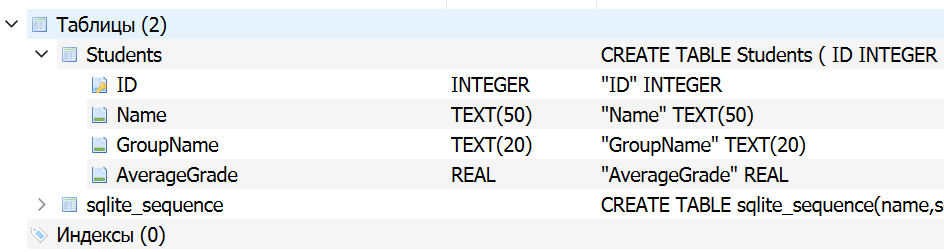


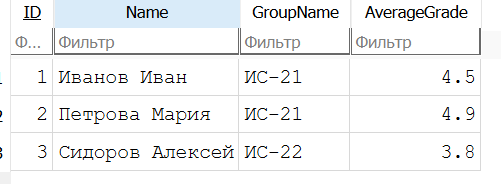
Создание таблицы Students



Добавление данных

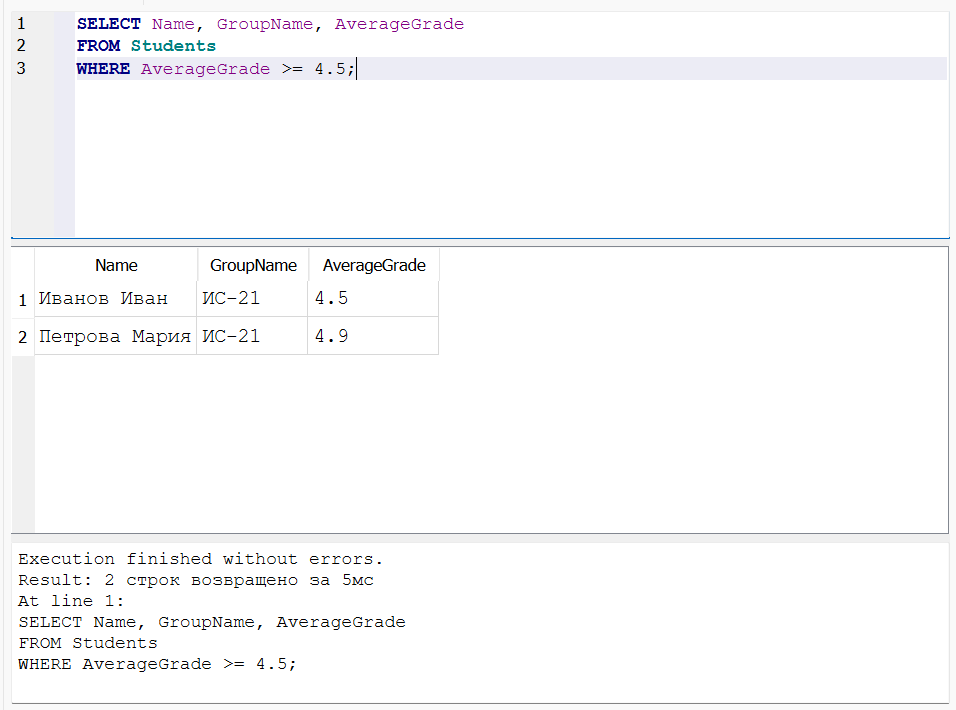






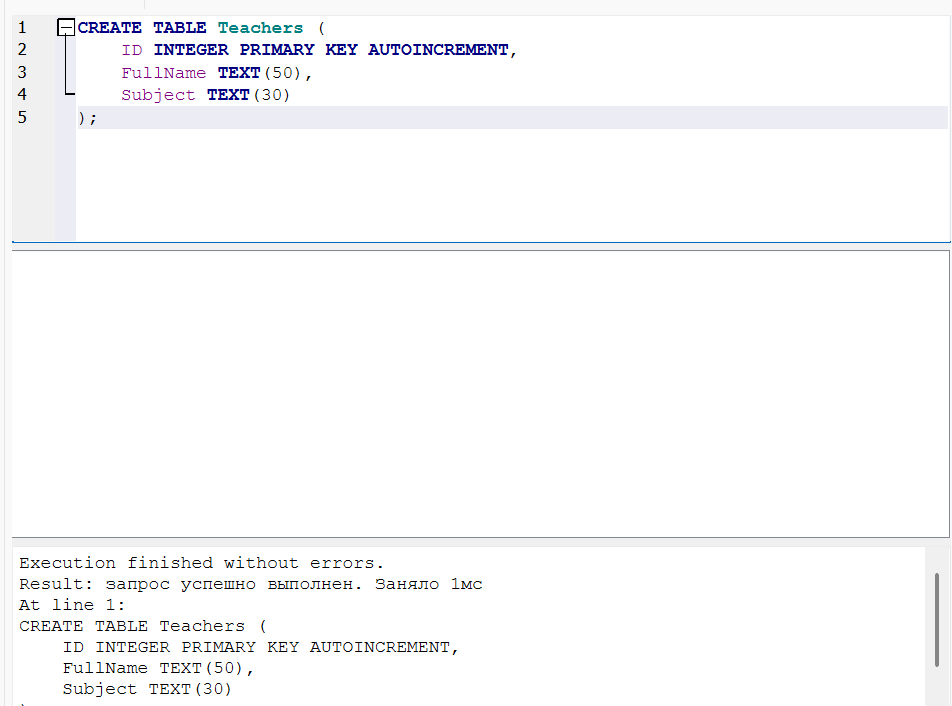
Создание файла SQL запросов

**МОДУЛЬ 1: STUDENTS**

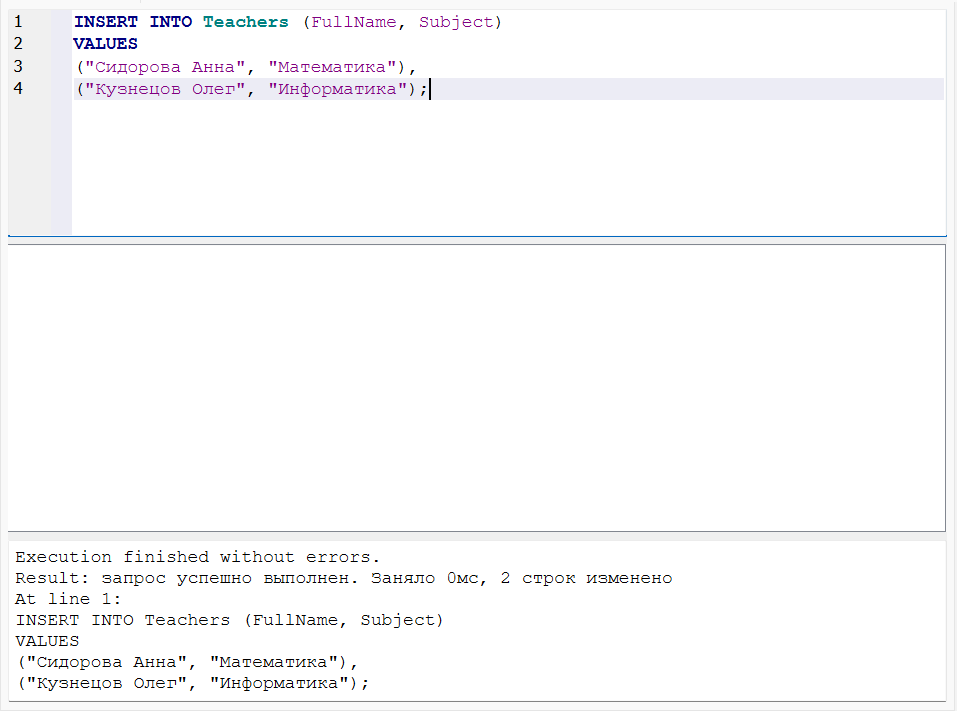


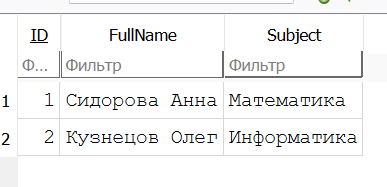
**МОДУЛЬ 2: TEACHERS**

Создание таблицы Teachers

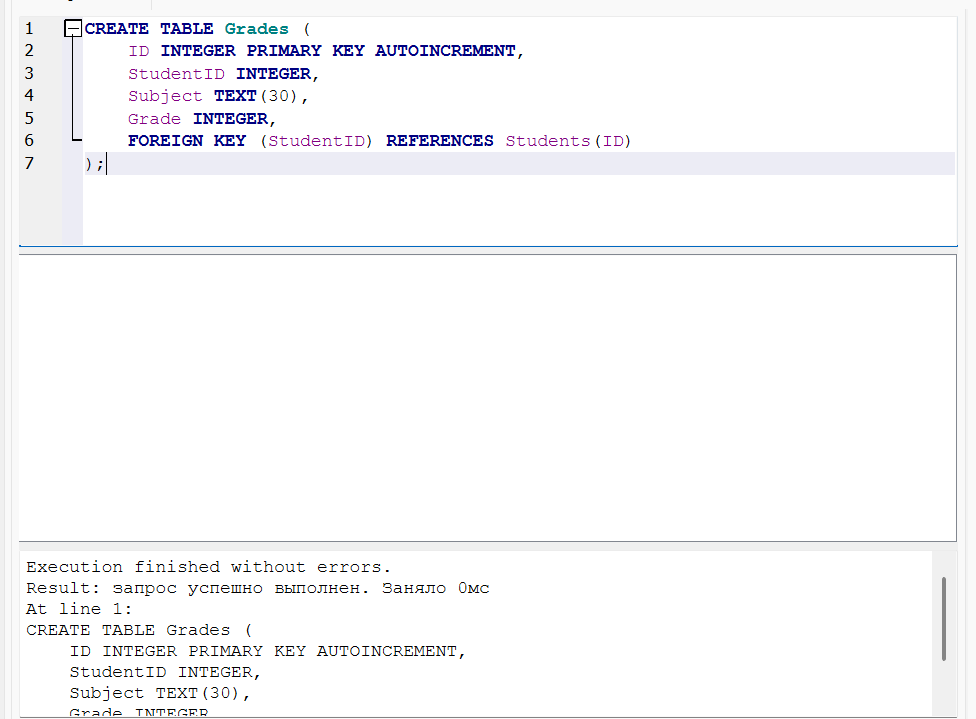


Добавление данных Teachers

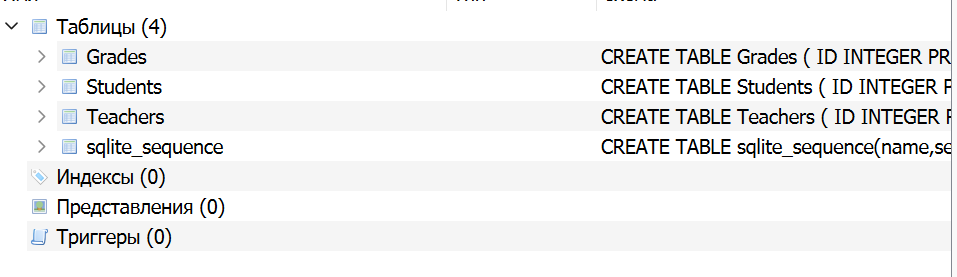


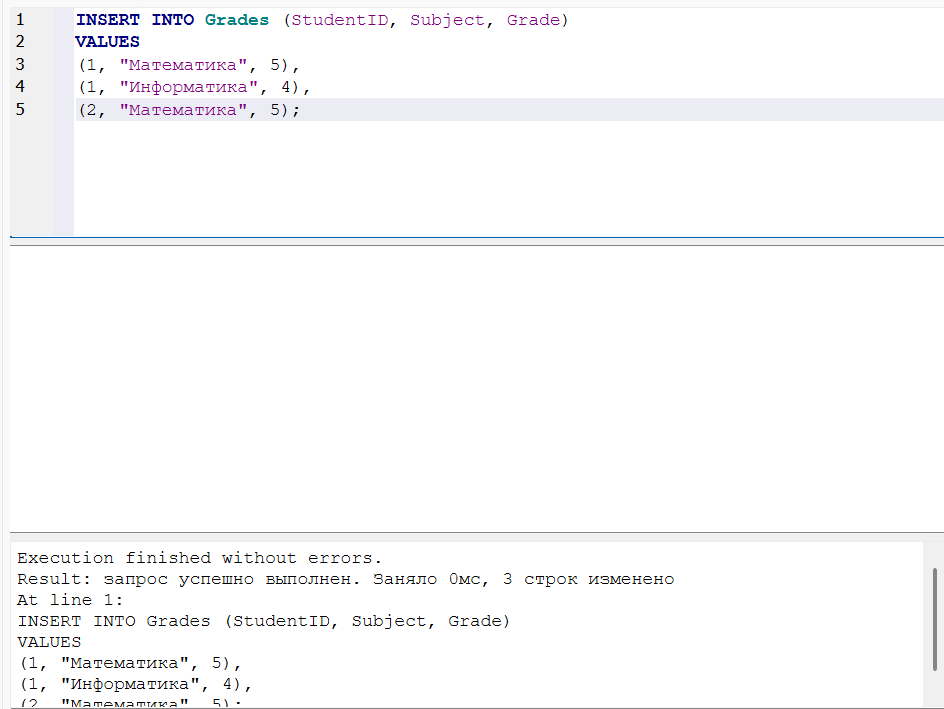


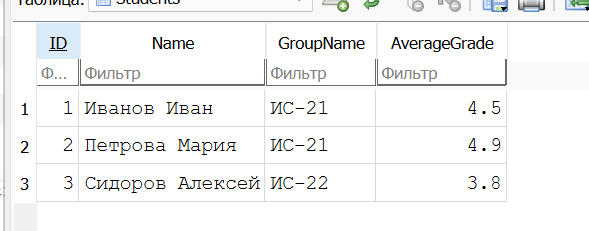
**МОДУЛЬ 3: GRADES**

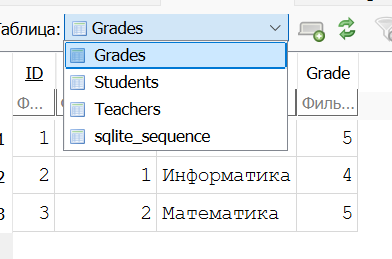


Добавление оценок

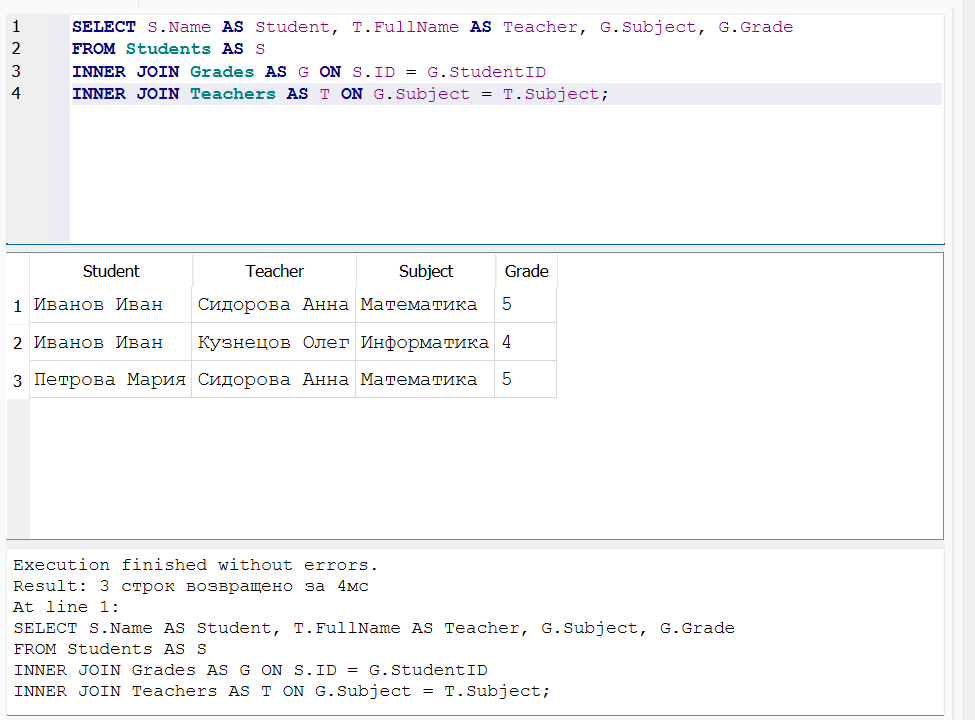


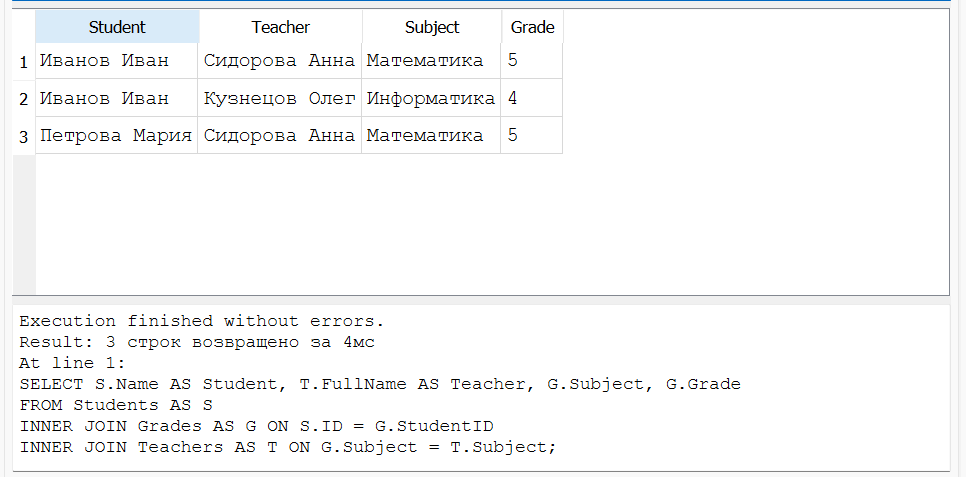




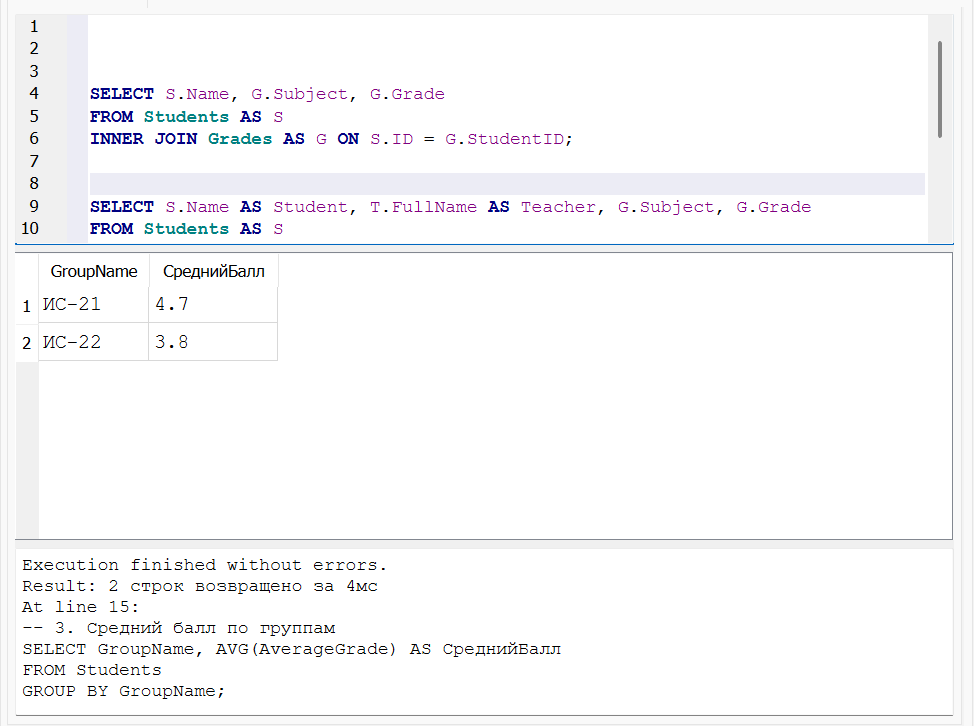


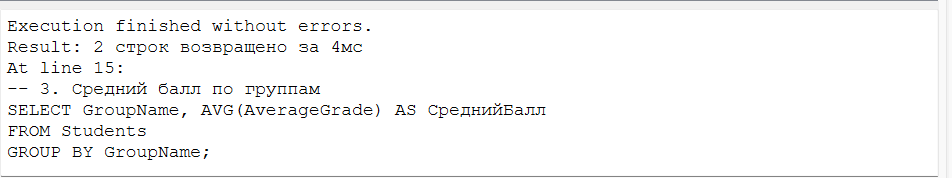
**СЛОЖНЫЙ ЗАПРОС**





Создание файла интегрированных запросов





SELECT S.Name, G.Subject, G.Grade

FROM Students AS S

INNER JOIN Grades AS G ON S.ID = G.StudentID;

SELECT S.Name AS Student, T.FullName AS Teacher, G.Subject, G.Grade

FROM Students AS S

INNER JOIN Grades AS G ON S.ID = G.StudentID

INNER JOIN Teachers AS T ON G.Subject = T.Subject

WHERE G.Grade >= 4;

SELECT GroupName, AVG(AverageGrade) AS СреднийБалл

FROM Students

GROUP BY GroupName;

**РАБОТА С GIT - КОМАНДНАЯ РАБОТА**

Создание веток для каждого разработчика

# Разработчик Students

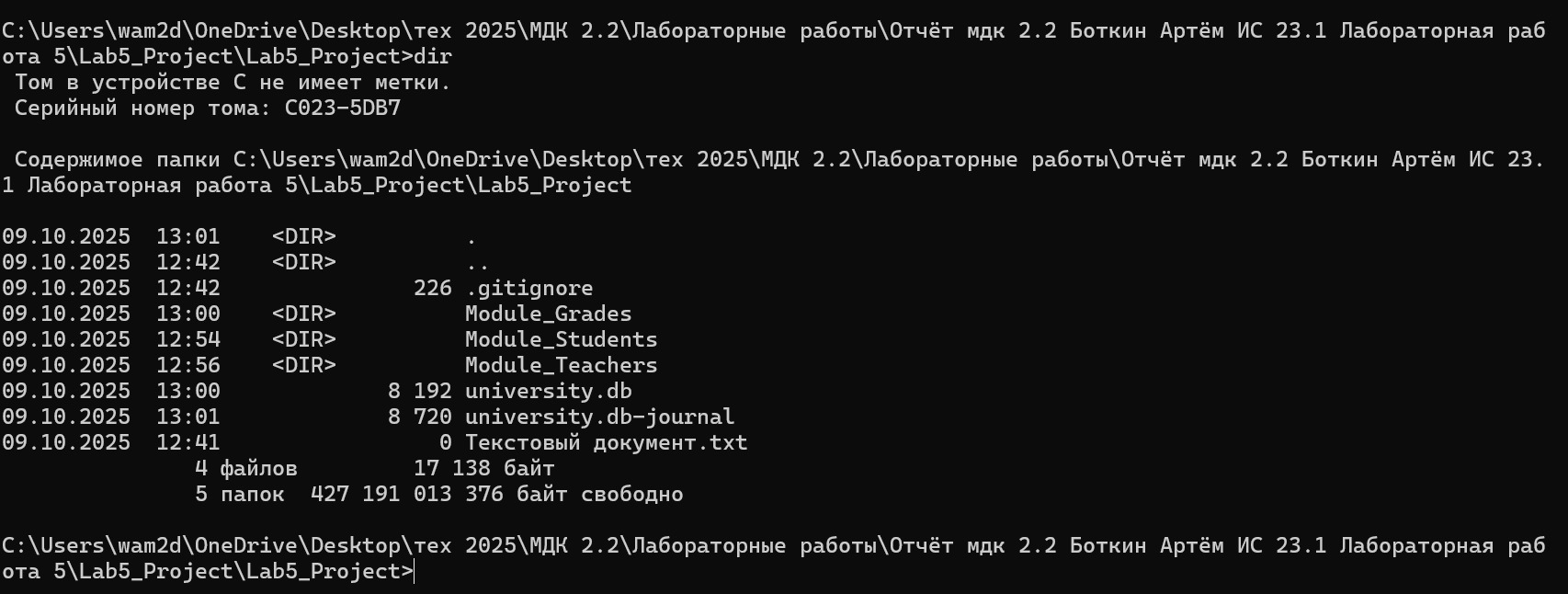
git checkout -b module-students

# Разработчик Teachers

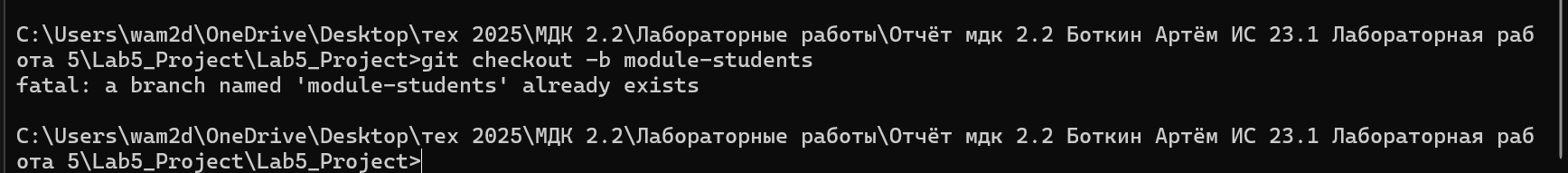
git checkout -b module-teachers

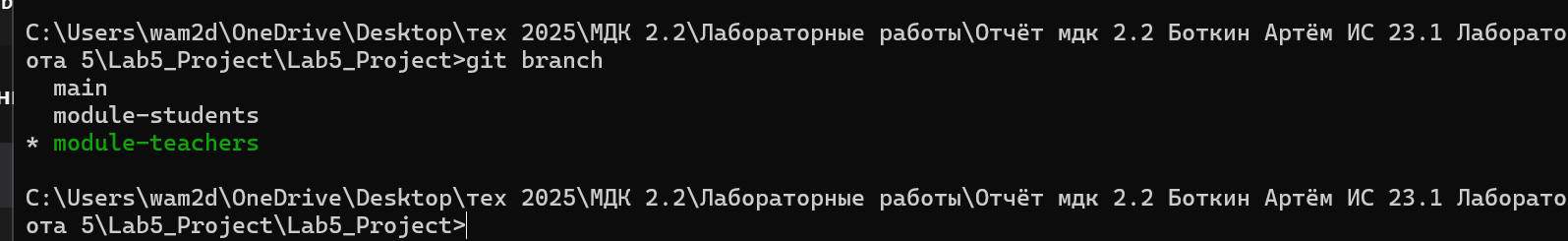
# Разработчик Grades

git checkout -b module-grades



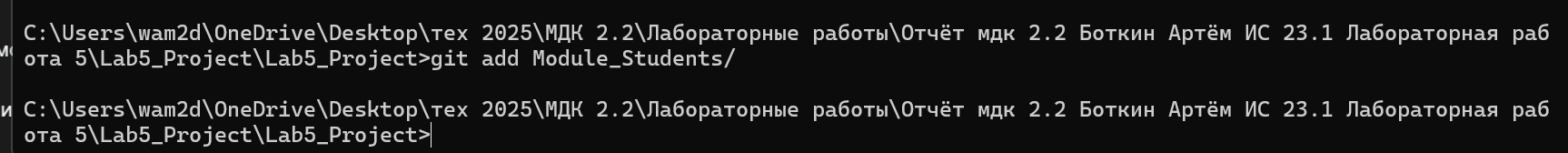
Создать ветку для модуля Students



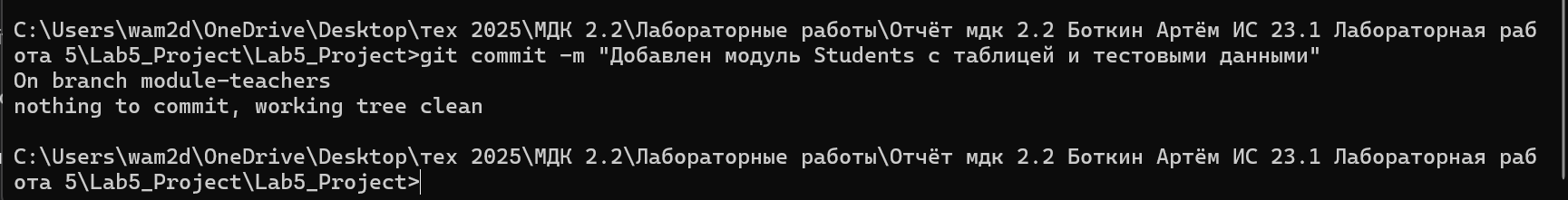


Добавить модуль Students в Git

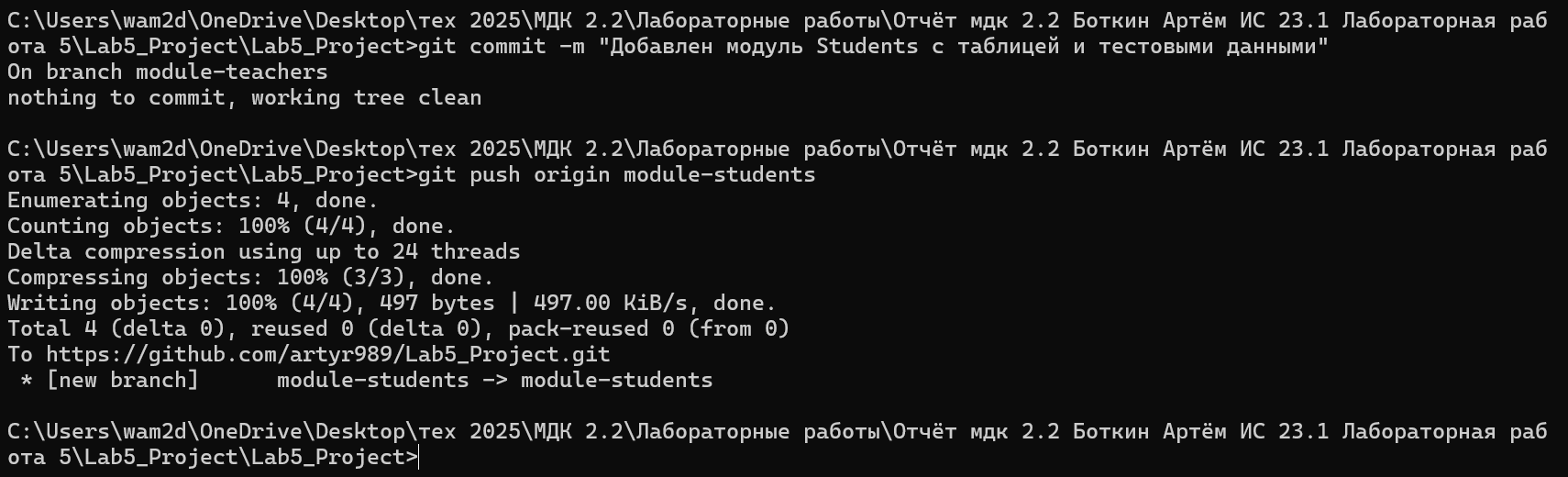
git add Module\_Students/



git commit -m "Добавлен модуль Students с таблицей и тестовыми данными"



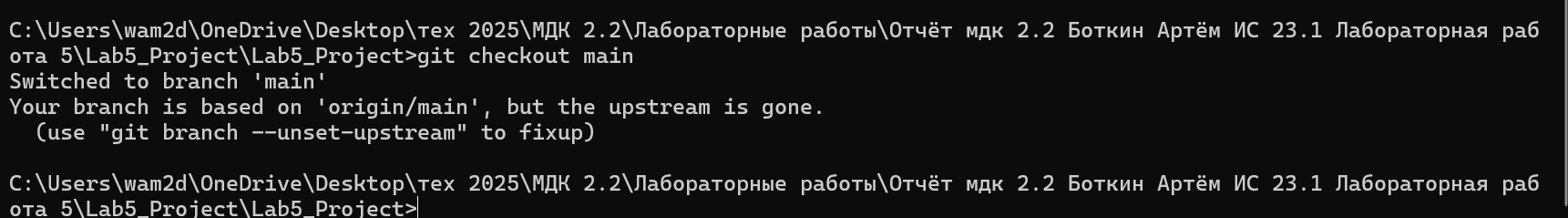
git push origin module-students



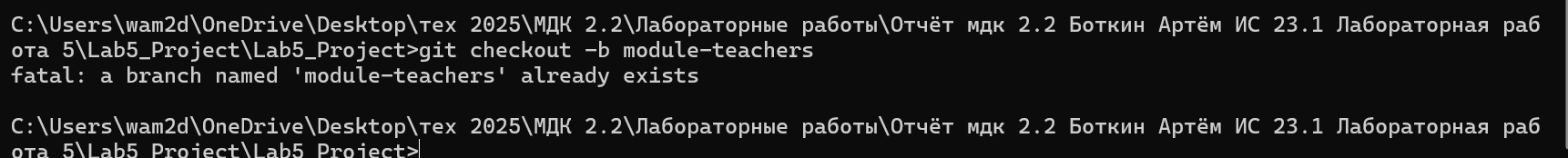
**ПЕРЕКЛЮЧИТЬСЯ НА МОДУЛЬ TEACHERS**

Вернуться на основную ветку main

git checkout main

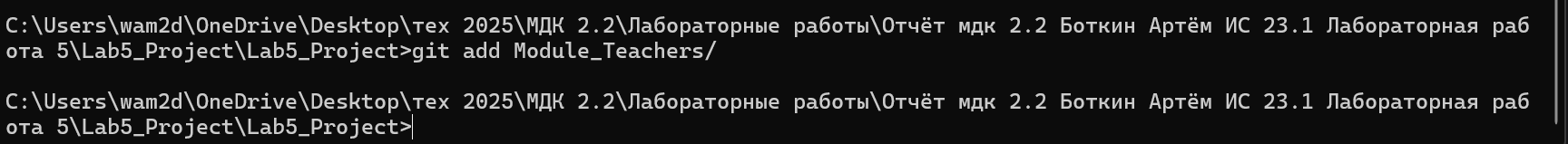


Создать ветку для модуля Teachers

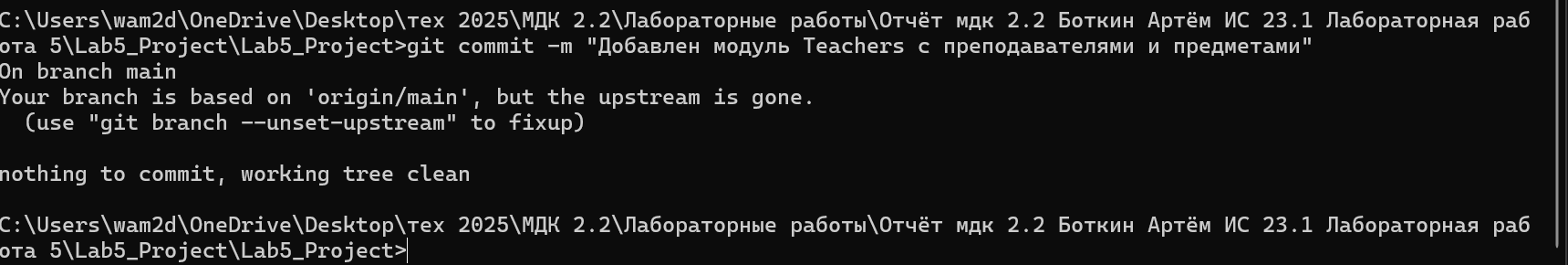


Добавить модуль Teachers

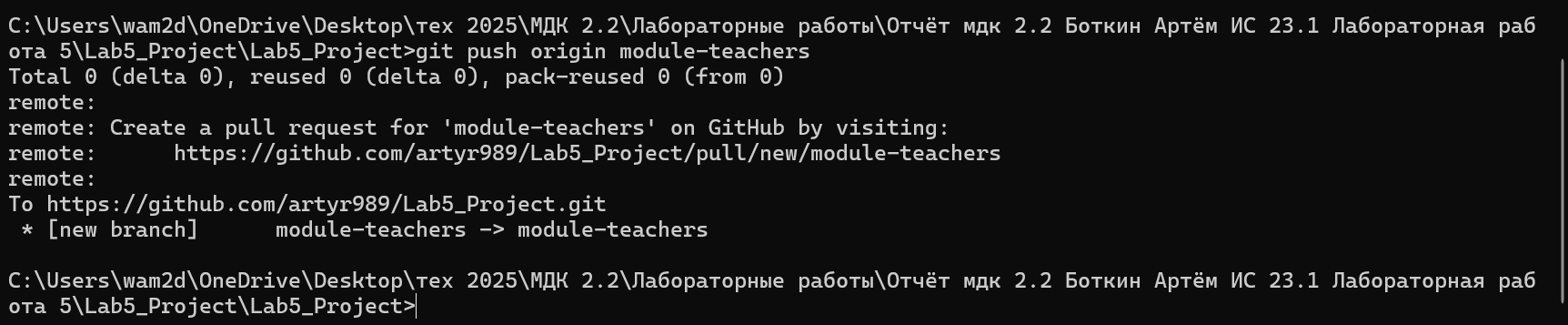
git add Module\_Teachers/



git commit -m "Добавлен модуль Teachers с преподавателями и предметами"



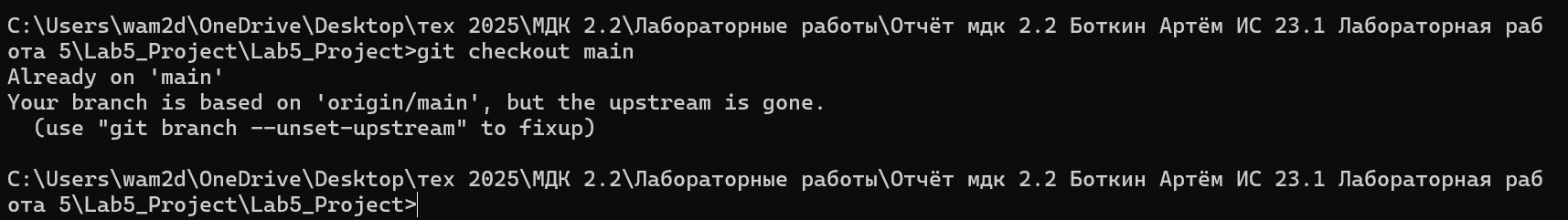
git push origin module-teachers



**ПЕРЕКЛЮЧИТЬСЯ НА МОДУЛЬ GRADES**

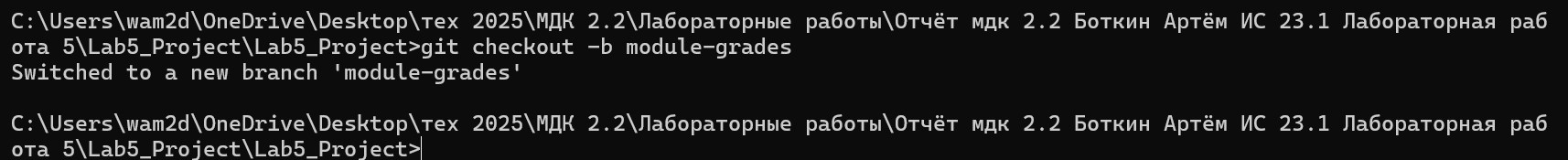
Вернуться на основную ветку

git checkout main



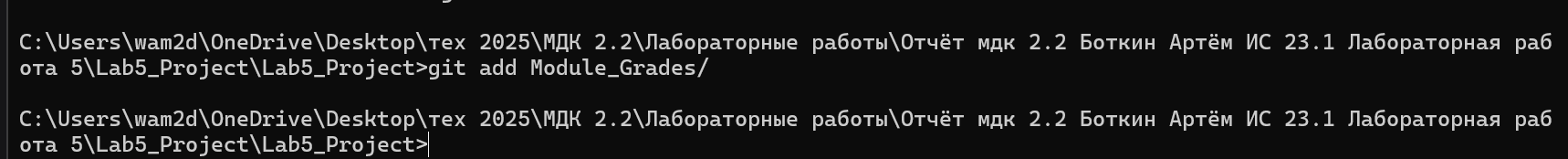
Создать ветку для модуля Grades

git checkout -b module-grades

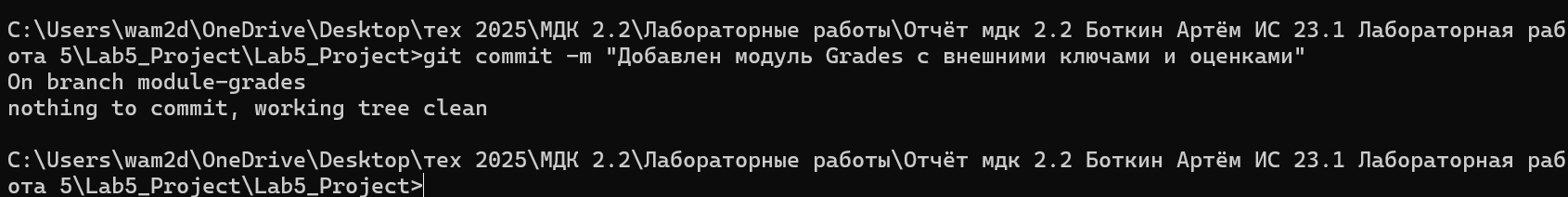


Добавить модуль Grades

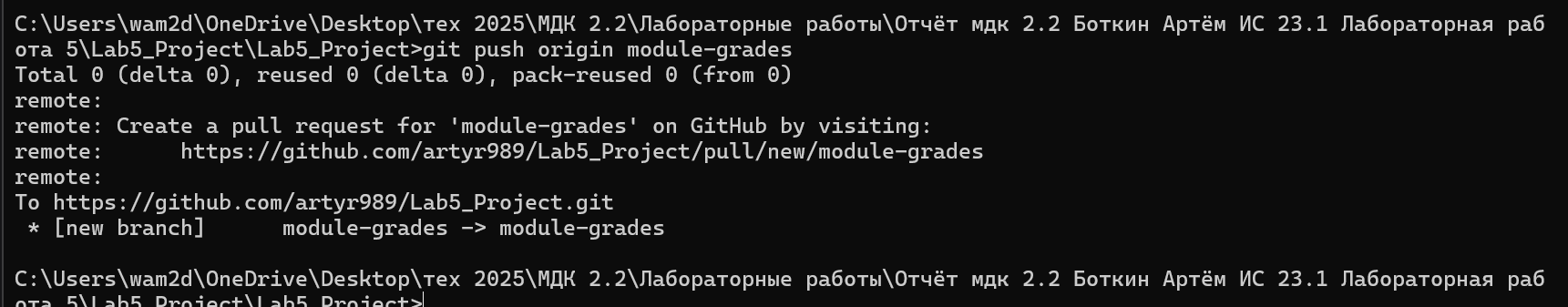
git add Module\_Grades/



git commit -m "Добавлен модуль Grades с внешними ключами и оценками"

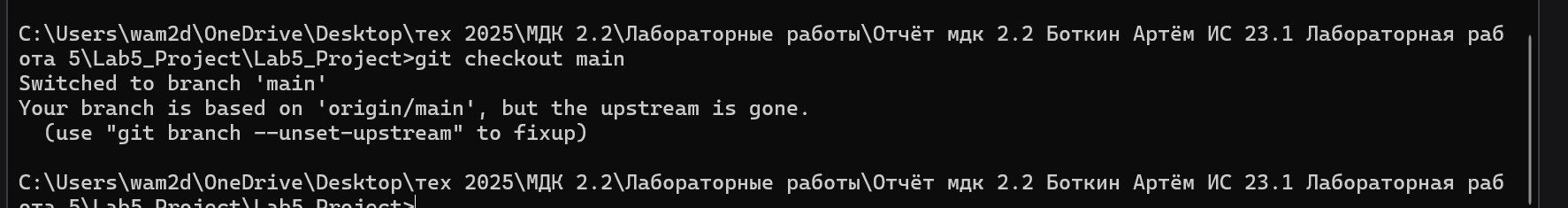


git push origin module-grades



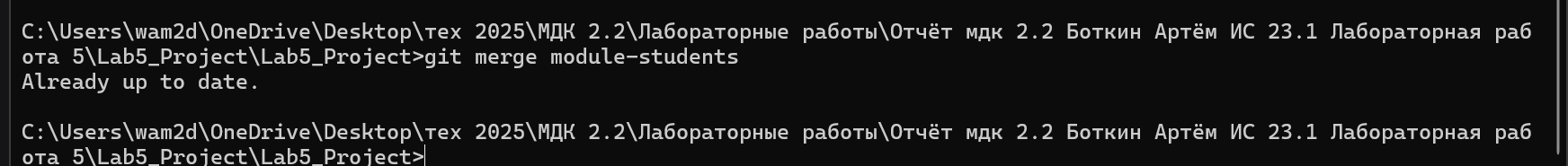
**ИНТЕГРАЦИЯ - СЛИЯНИЕ ВСЕХ ВЕТОК**

ВЕРНУТЬСЯ НА ОСНОВНУЮ ВЕТКУ



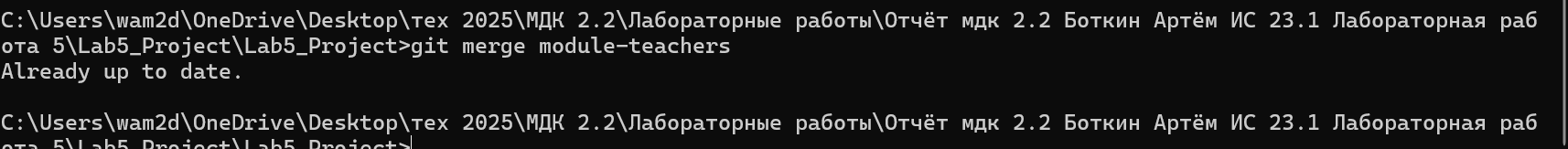
СЛИТЬ ВСЕ ВЕТКИ В ОДНУ

Слить ветку module-students



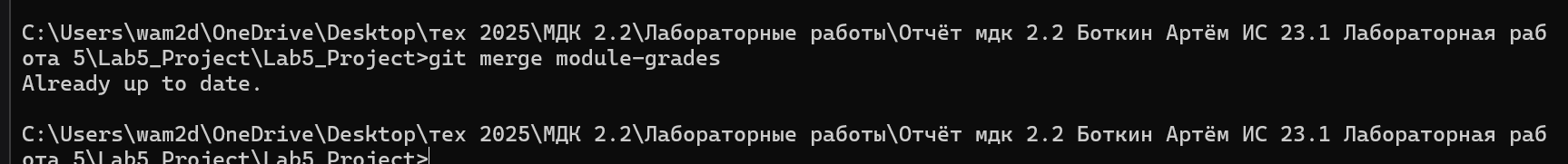
Слить ветку module-teachers

git merge module-teachers



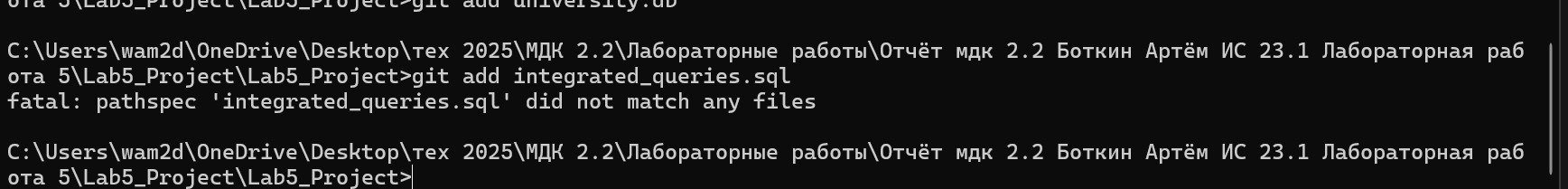
Слить ветку module-grades

git merge module-grades

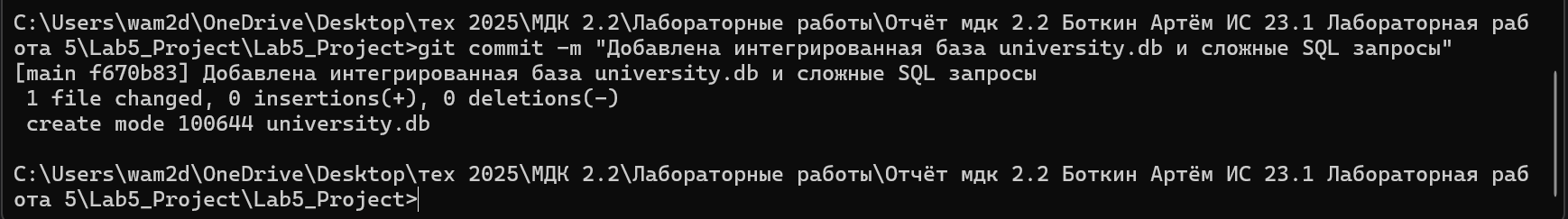


**ДОБАВИТЬ ИНТЕГРИРОВАННУЮ БАЗУ ДАННЫХ**

git add university.db

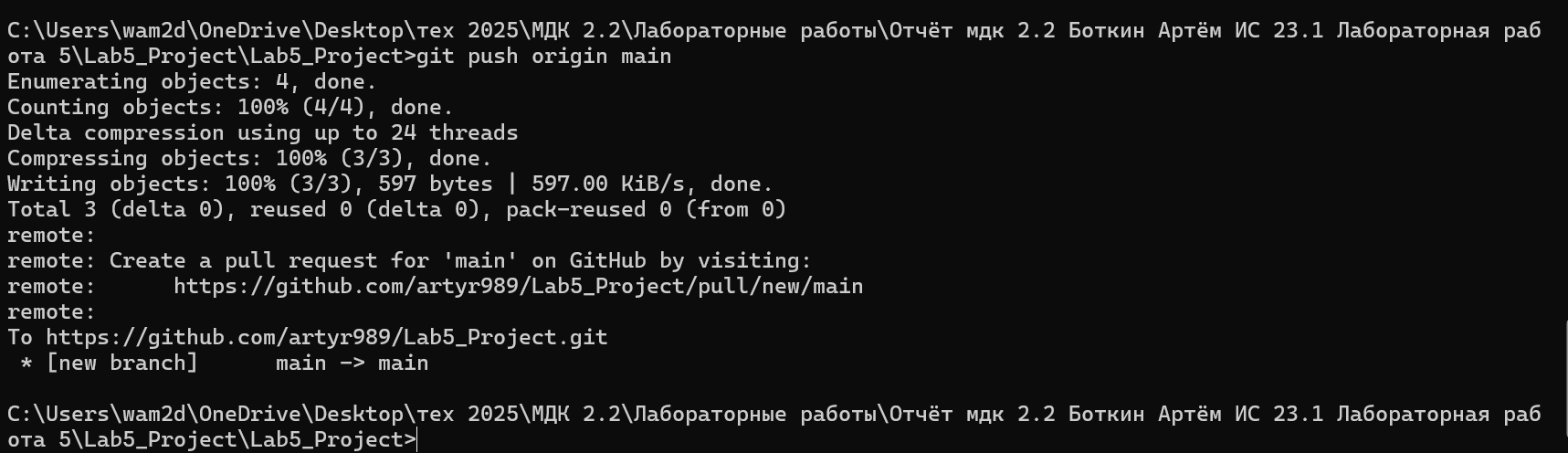


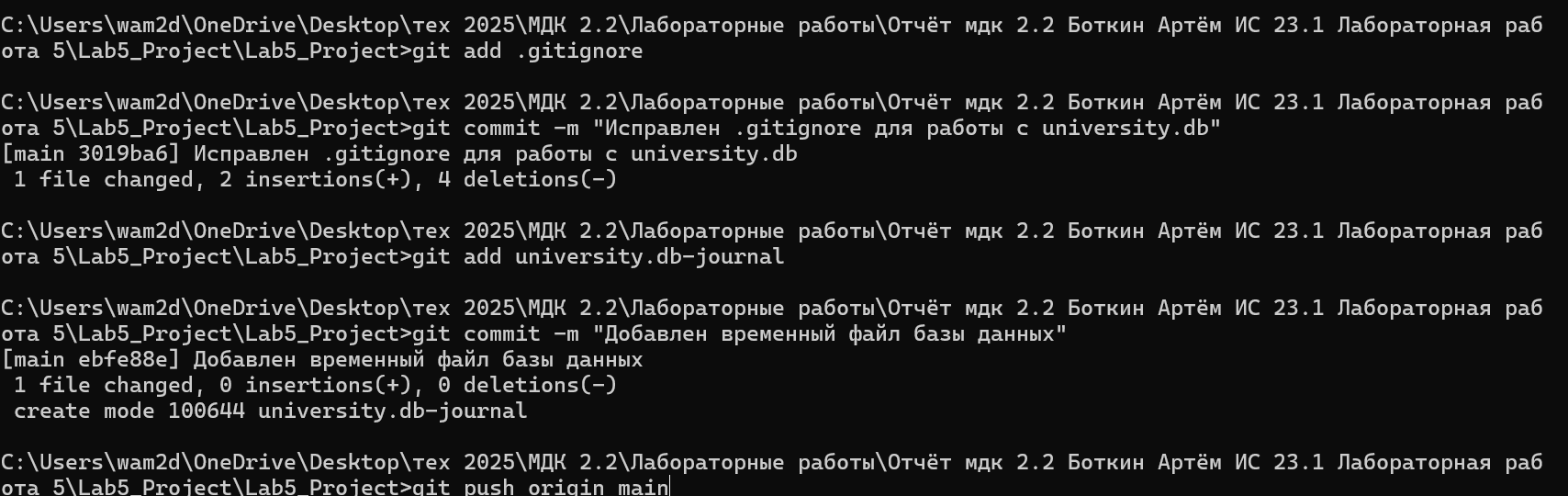
git commit -m "Добавлена интегрированная база university.db и сложные SQL запросы"

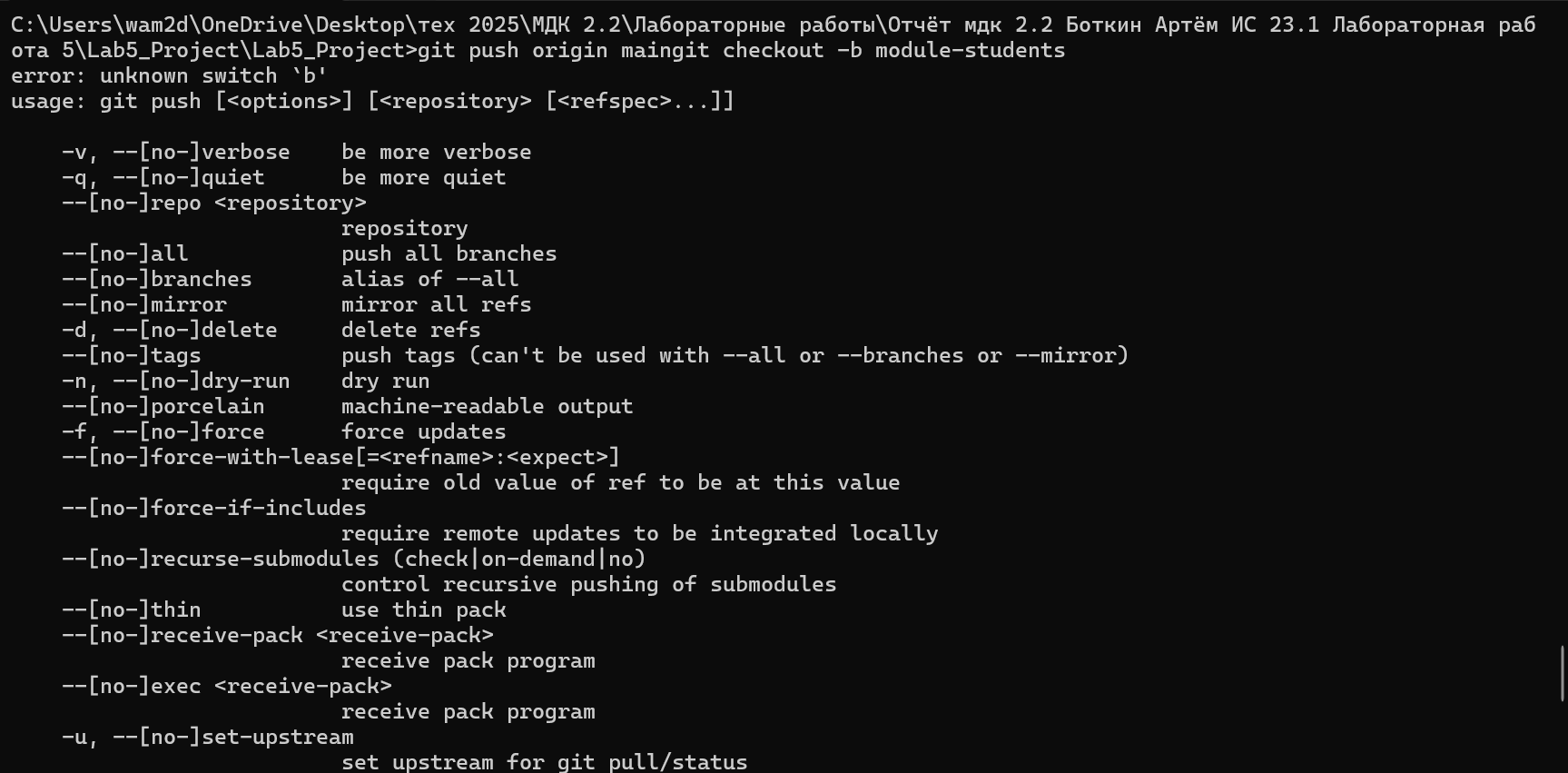


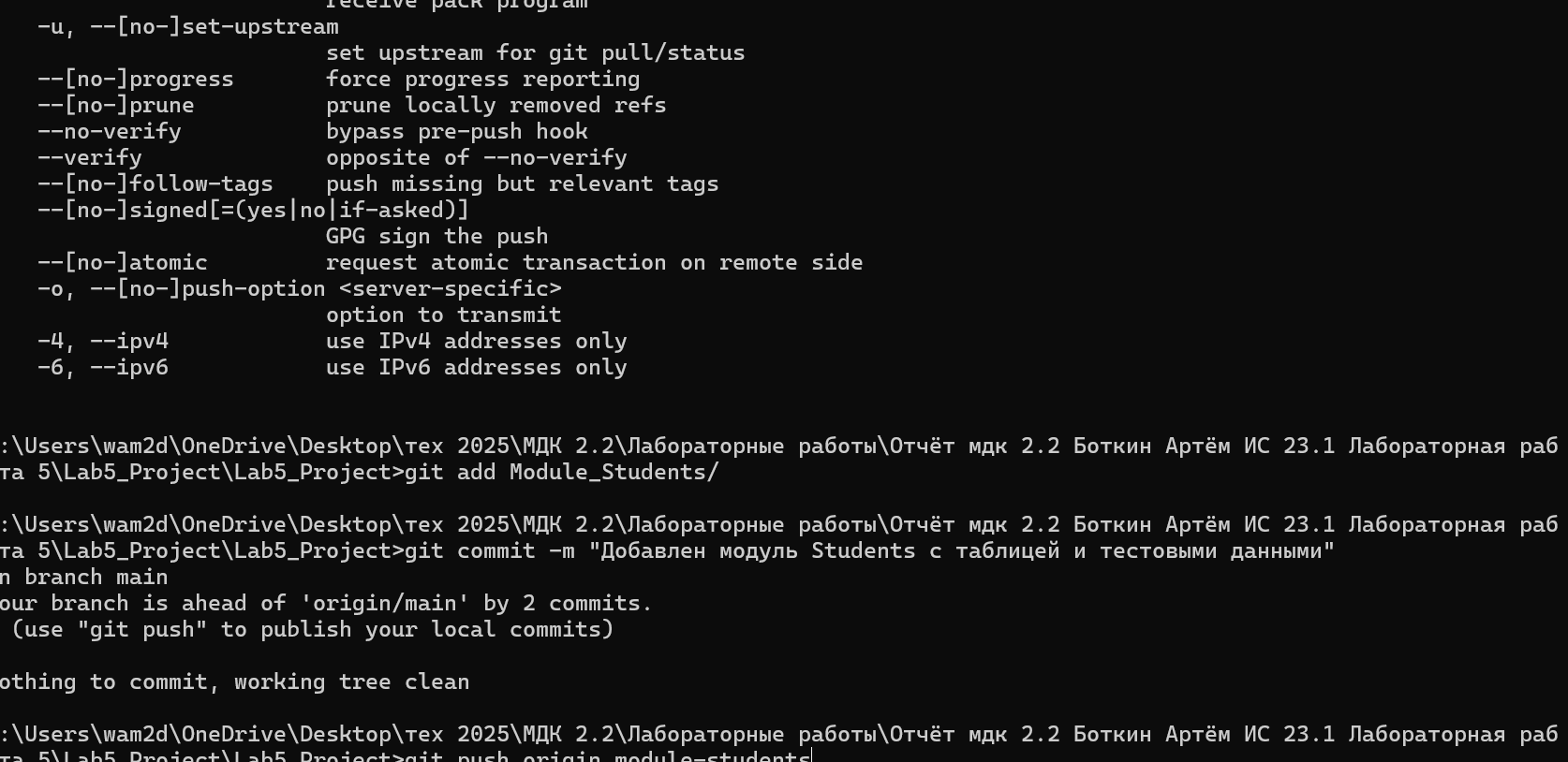
**ОТПРАВИТЬ ВСЕ ИЗМЕНЕНИЯ НА GITHUB**

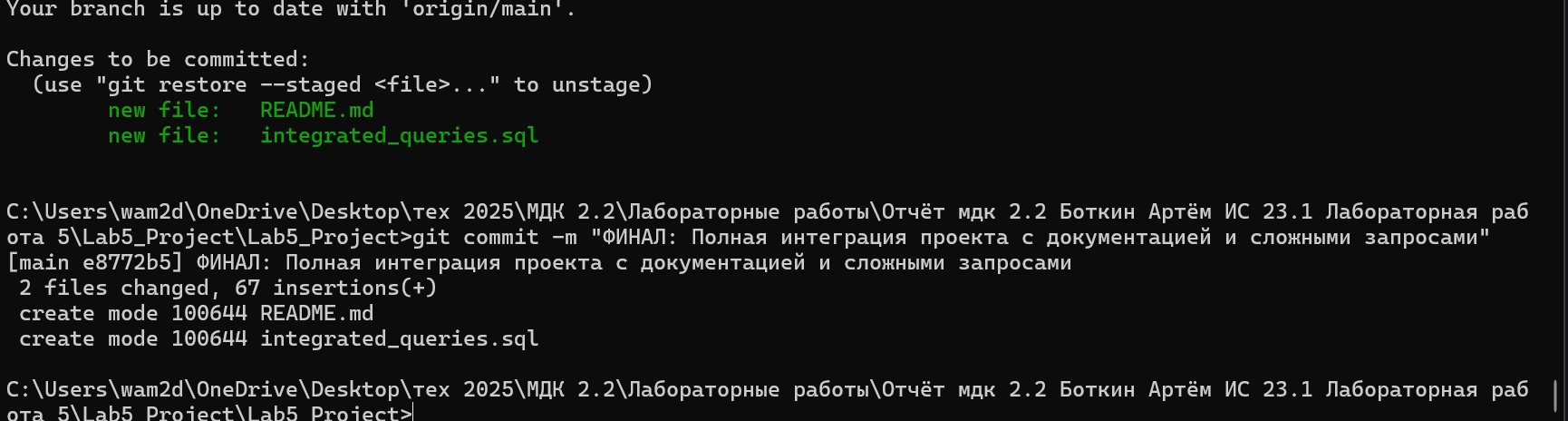
git push origin main

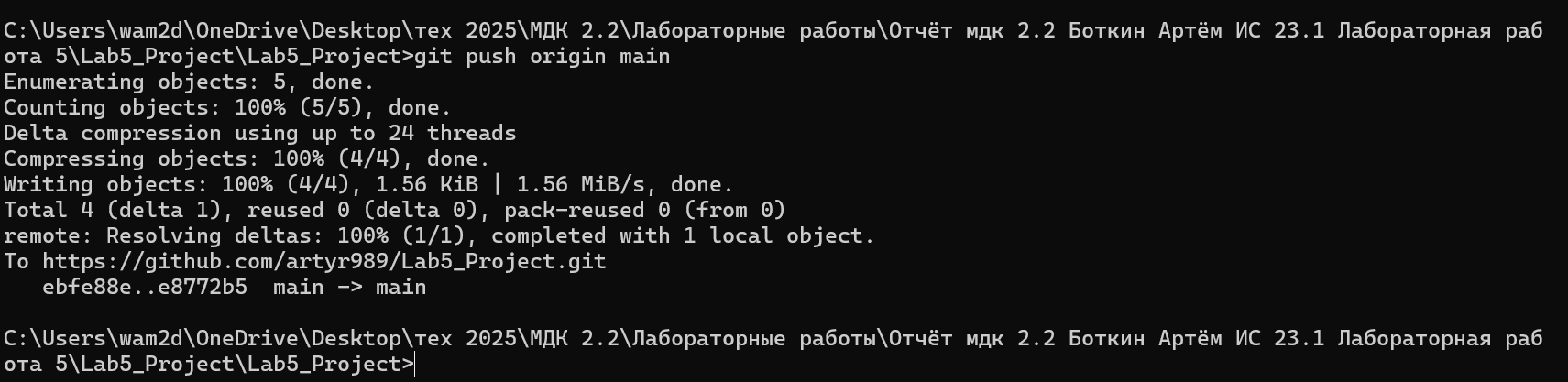
****

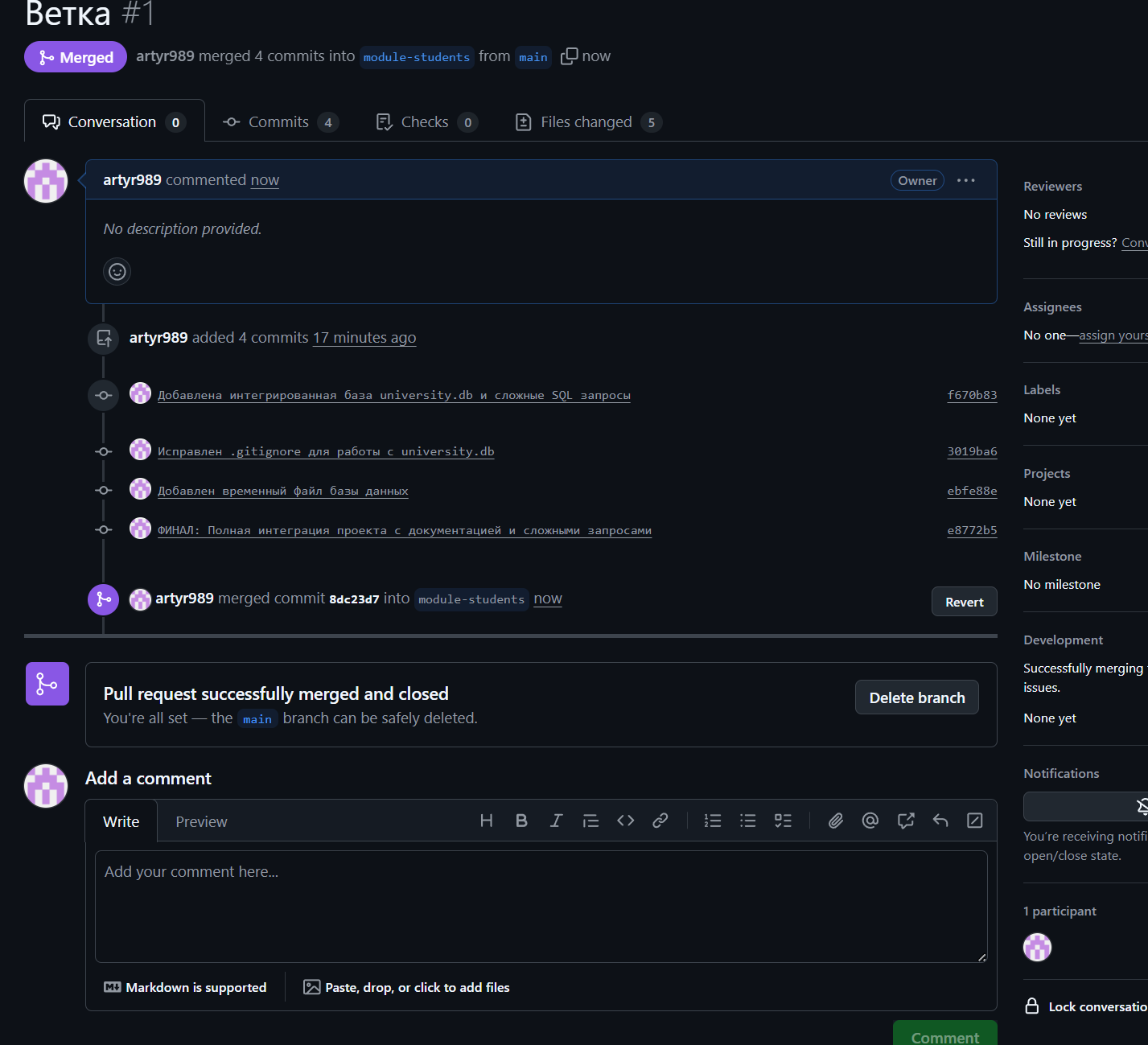
****

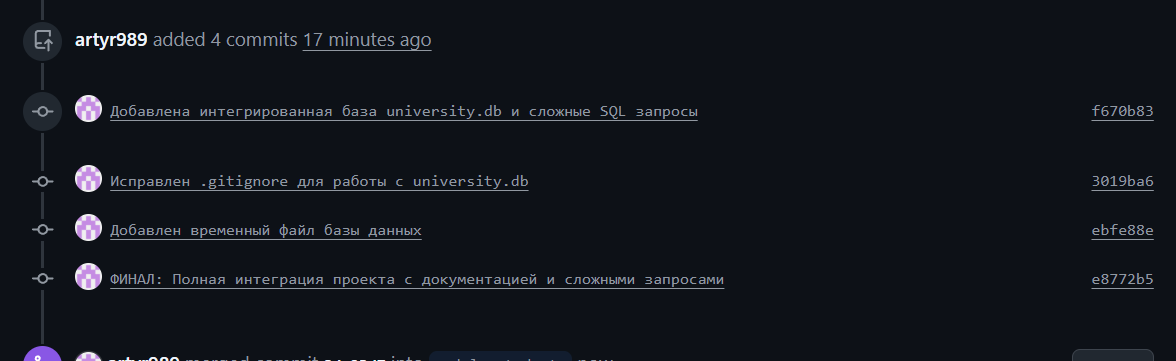
****

****

****

****

****



**Контрольные вопросы для защиты**

**1. Что такое модуль проекта?**

Это независимая часть системы с конкретной задачей. Например, модуль "Студенты" отвечает за данные студентов, "Преподаватели" — за преподавателей.

**2. Как открыть SQL-режим в MS Access?**

Вкладка "Создание" → "Конструктор запросов" → "Режим SQL".

**3. Для чего используется .gitignore?**

Чтобы Git игнорировал временные и служебные файлы (например, \*.db, \*.tmp), не добавляя их в репозиторий.

**4. Как создать и переключиться на ветку в Git?**

git checkout -b имя\_ветки — создает и переключается.

git checkout имя\_ветки — переключается на существующую.

**5. Что такое коммит и зачем он нужен?**

Это сохранение текущей версии проекта. Нужен для фиксации изменений и возможности отката.

**6. Как выполняется слияние веток?**

Переключиться на основную ветку: git checkout main, затем git merge имя\_ветки.

**7. Как разрешить конфликт при merge?**

Вручную редактируем файлы, устраняя конфликтные участки, потом делаем коммит.

8. Пример запроса SELECT:

sql

SELECT Name, GroupName FROM Students WHERE AverageGrade >= 4.5;

9. Пример запроса INSERT:

sql

INSERT INTO Students (Name, GroupName) VALUES ('Иванов Иван', 'ИС-21');

10. Как проверить связь между таблицами?

Через JOIN-запрос:

sql

SELECT S.Name, G.Grade

FROM Students S

JOIN Grades G ON S.ID = G.StudentID;

11. Для чего нужен README.md?

Это документация проекта: описание, инструкции по запуску, структура базы.

**12. Как подсчитать средний балл по предмету?**

sql

SELECT AVG(Grade) FROM Grades WHERE Subject = 'Математика';

**13. Чем отличается INNER JOIN от LEFT JOIN?**

INNER JOIN показывает только совпадающие записи, LEFT JOIN — все из левой таблицы, даже без совпадений.

**14. Как тестировать SQL-запросы в Access?**

Через "Режим SQL" — вводим запрос и выполняем, смотрим результат.

**15. Почему важно распределять работу по модулям в команде?**

Чтобы работать параллельно, не мешая друг другу, и упростить тестирование и интеграцию.