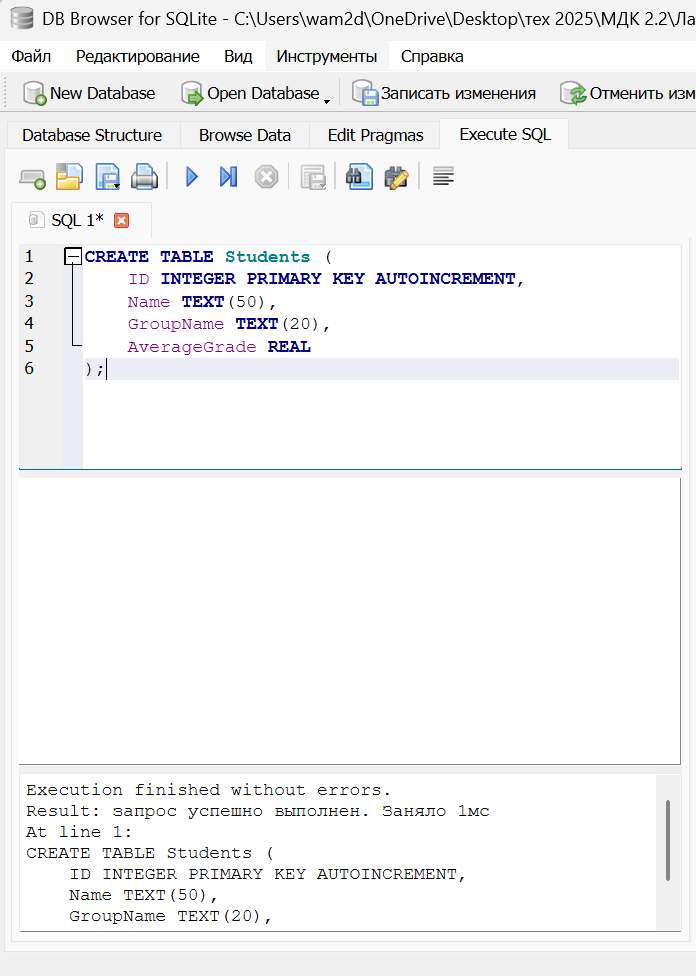
Лабораторная работа 4

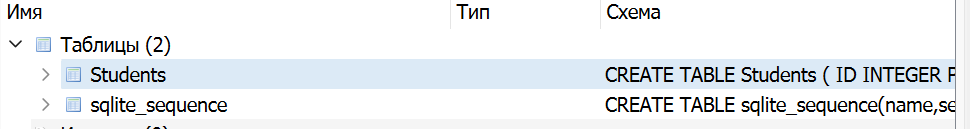
**Лабораторная работам 4** «Настройка работы системы контроля версий (типов импортируемых файлов, путей, фильтров и др. параметров импорта в репозиторий)»

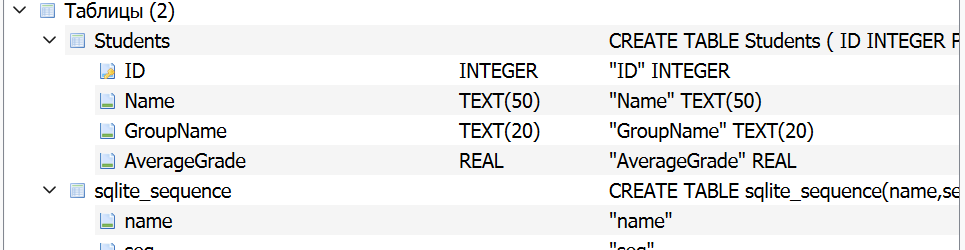
**Цель работы**

Освоить навыки работы с системой контроля версий Git при организации проектов баз данных в Microsoft Access.  
Научиться работать с SQL-запросами в MS Access, выполнять импорт, настройку фильтров, путей и параметров хранения файлов в репозитории Git.

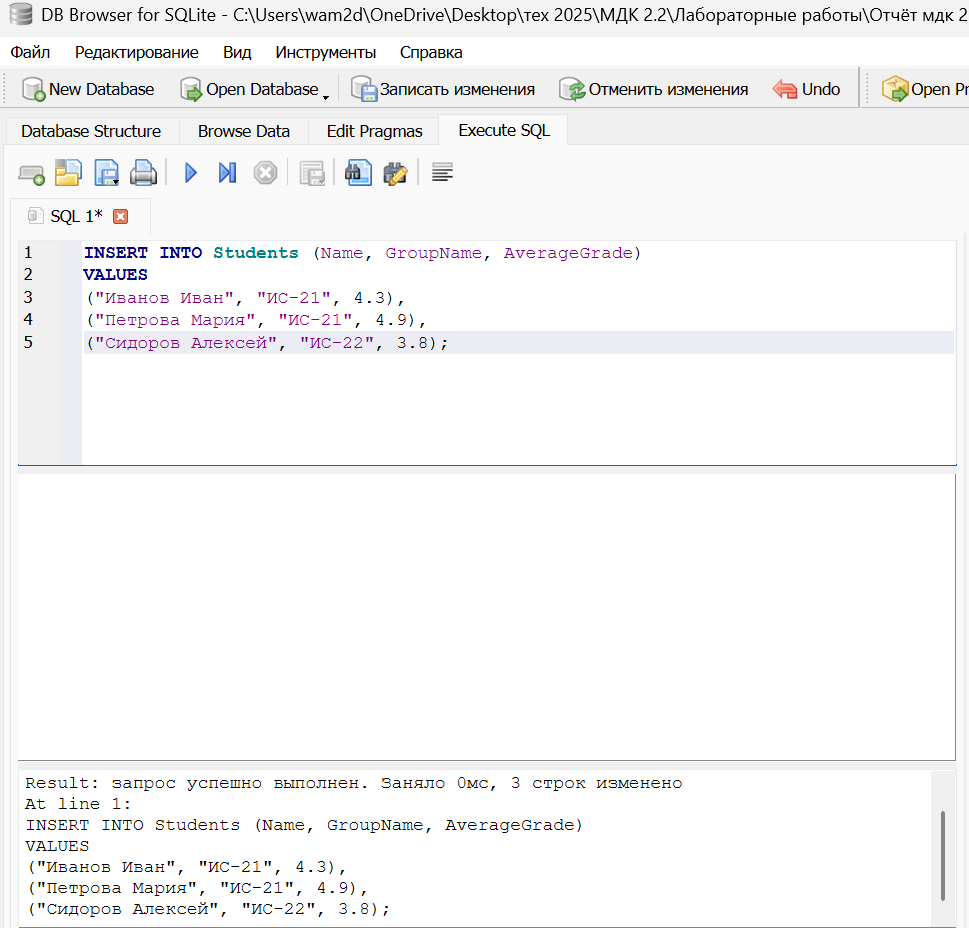
**Создание таблицы Students**

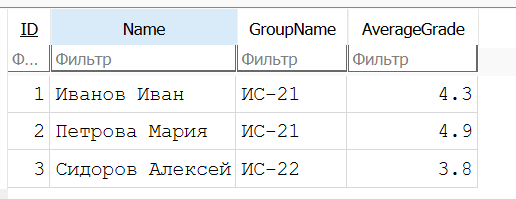
****

****

****

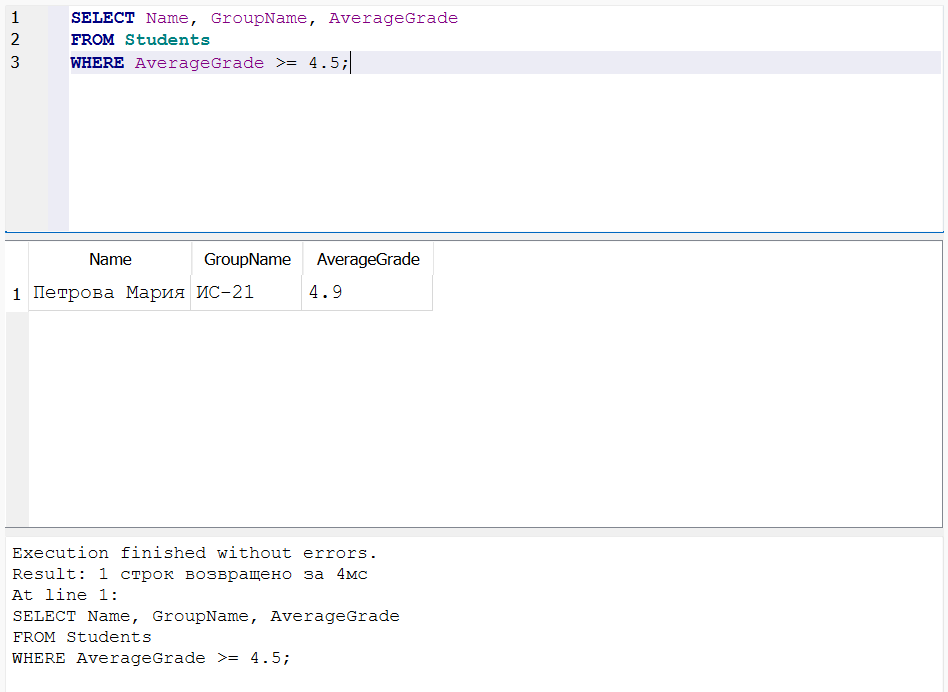
**Работа с SQL-запросами**

****

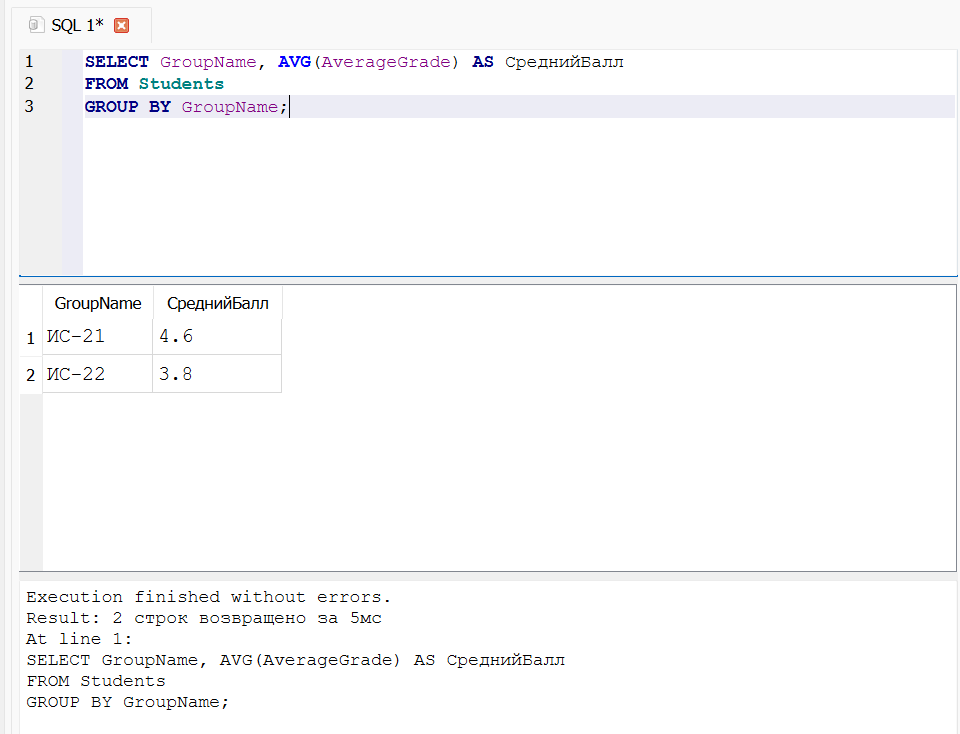
****

**Выполнение SQL-запросов**

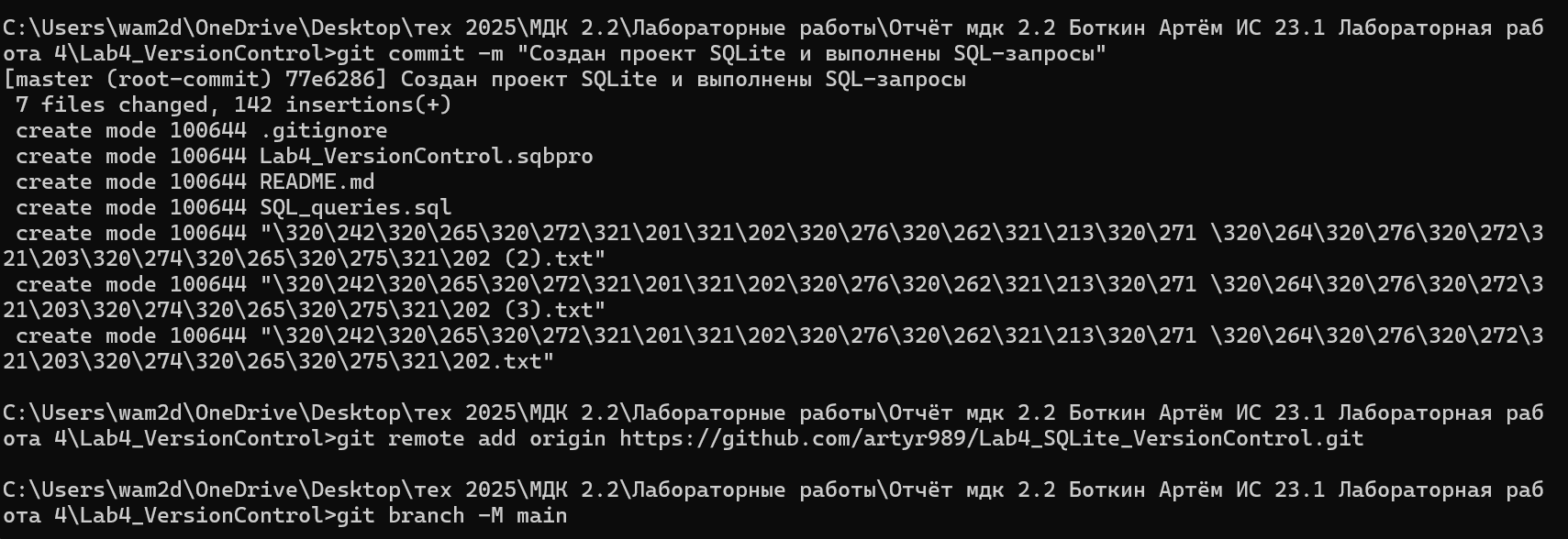
Запрос 1: Выборка отличников

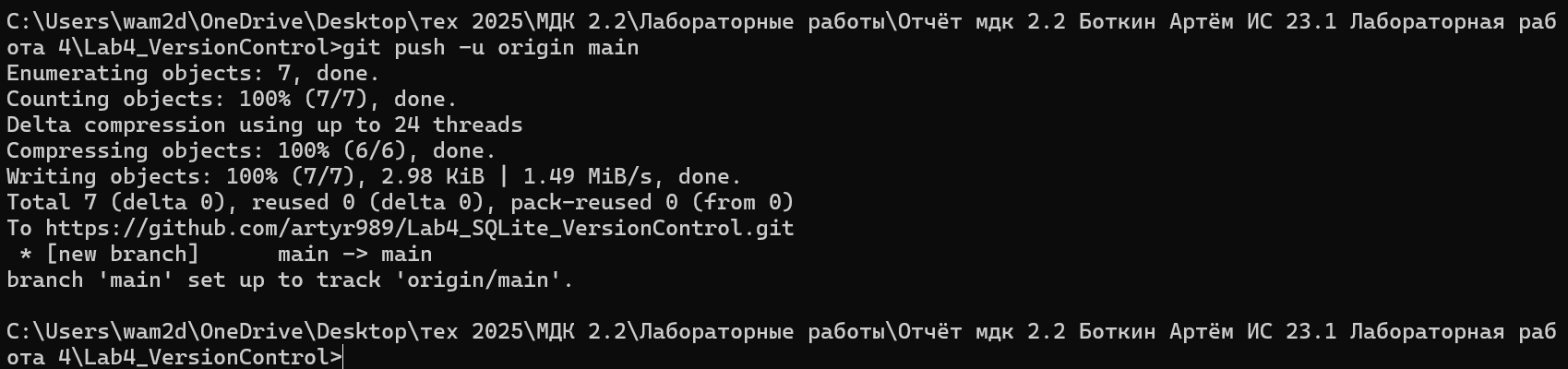


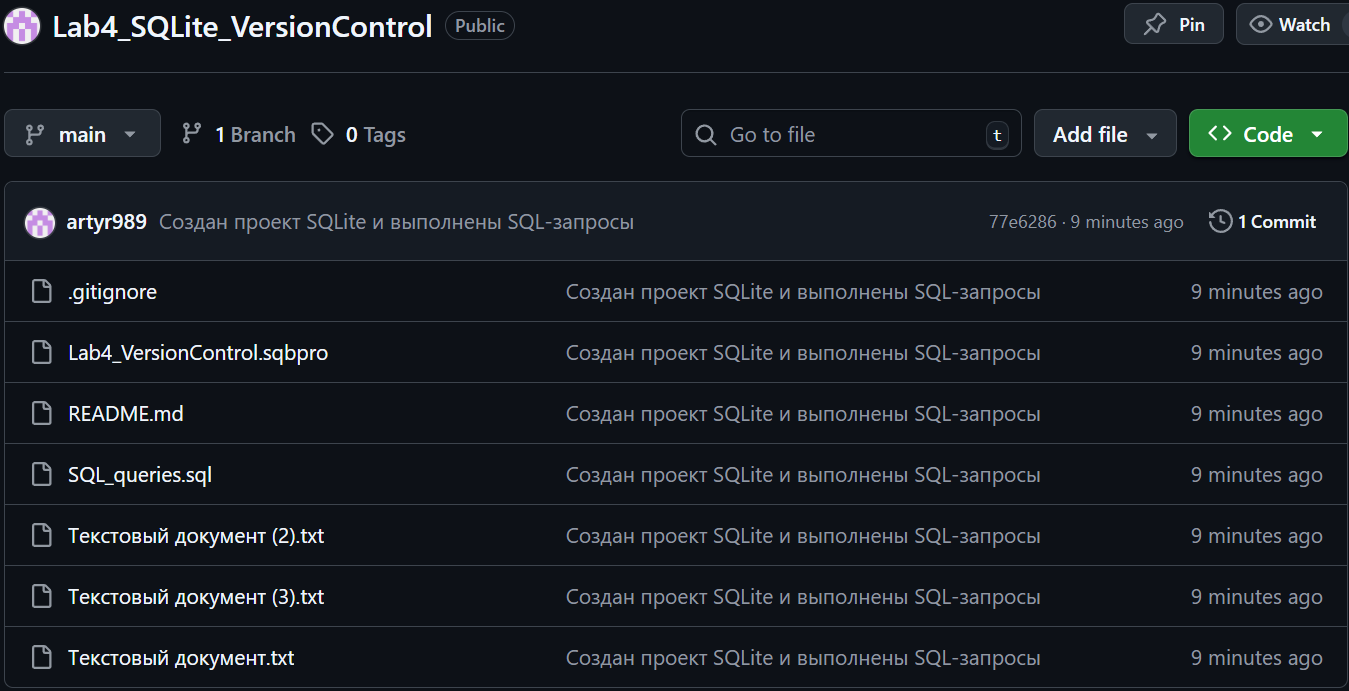
Запрос 2: Агрегирование по группам

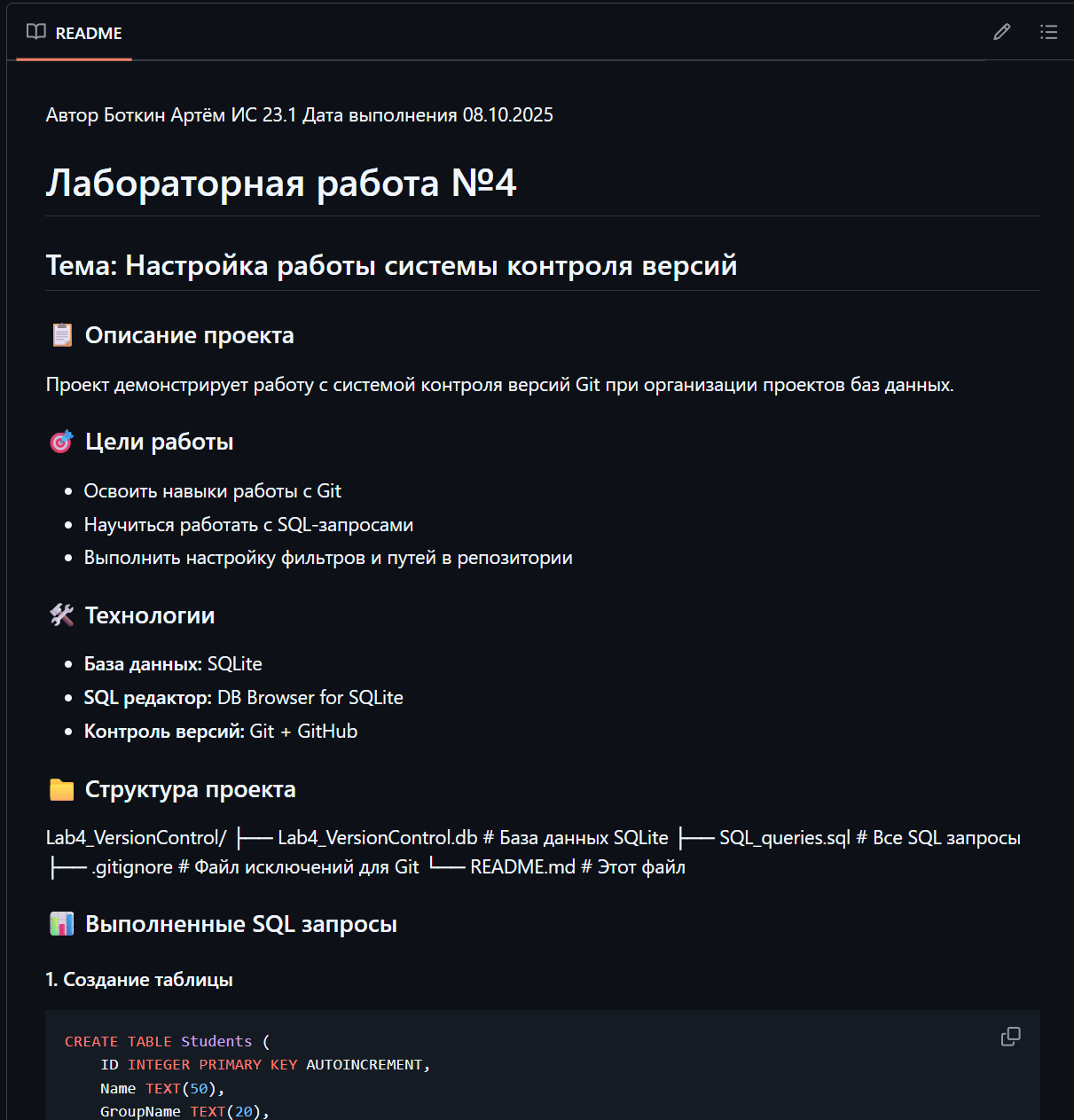


**РАБОТА С GIT И GITHUB**







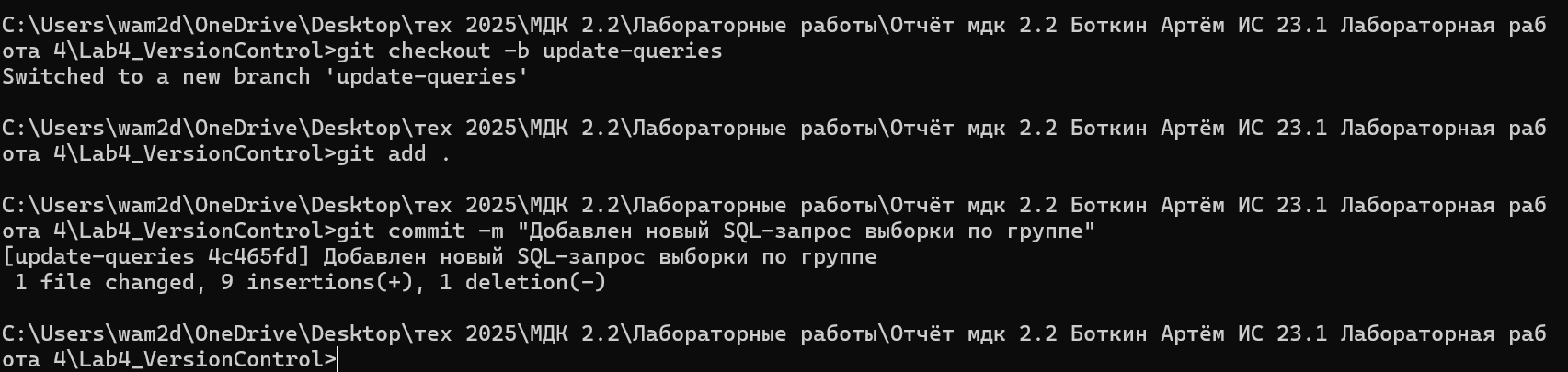


**СОЗДАНИЕ ВЕТКИ И PULL REQUEST**

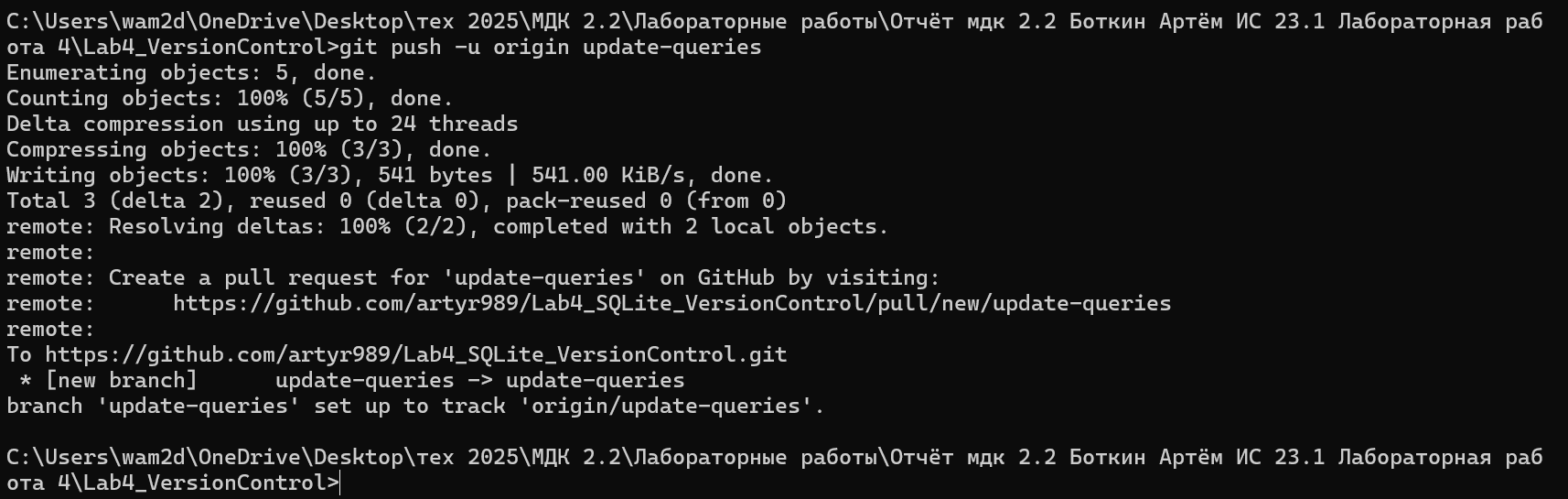
Создание новой ветки-git checkout -b update-queries

Сохранение изменений в новой ветке- git add .

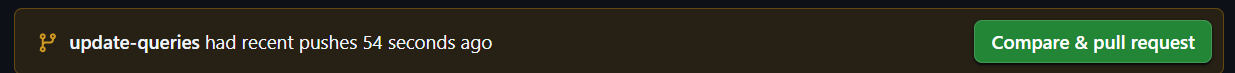
git commit -m "Добавлен новый SQL-запрос выборки по группе"

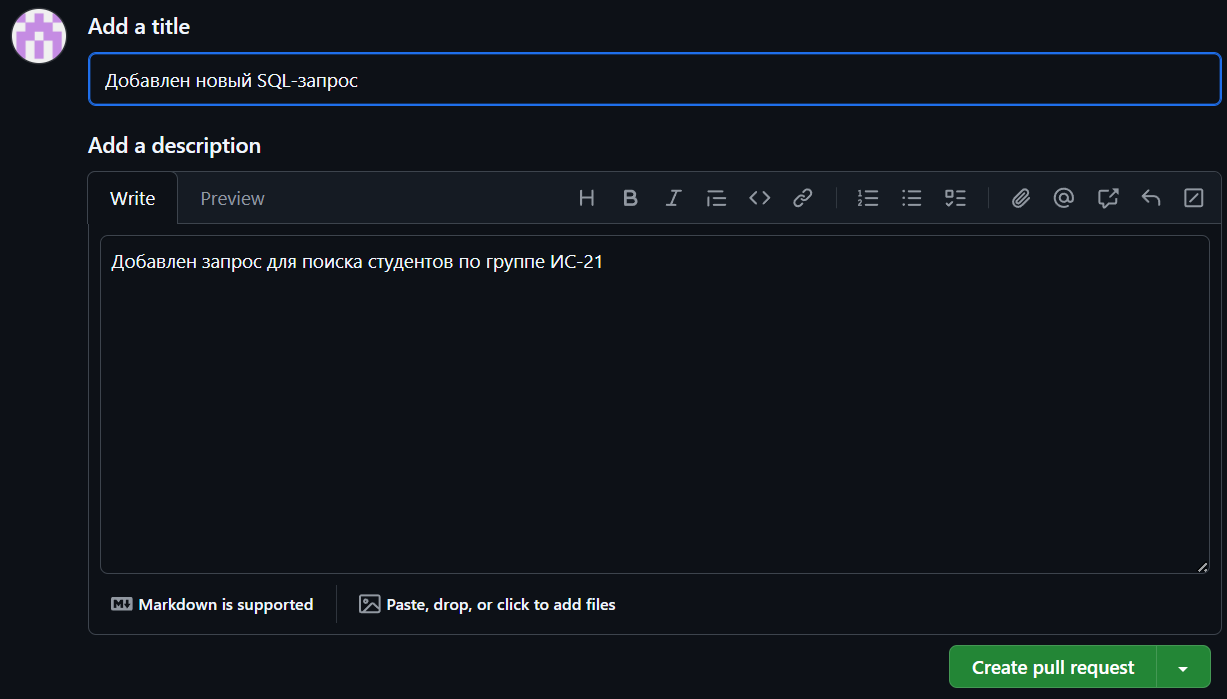


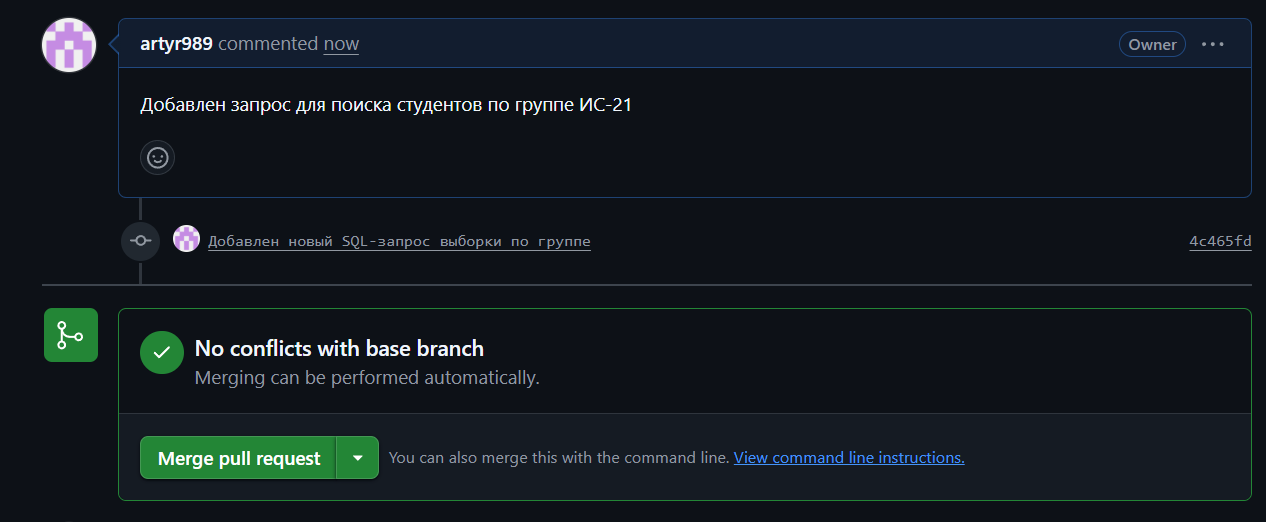
Отправка новой ветки на GitHub - git push -u origin update-queries

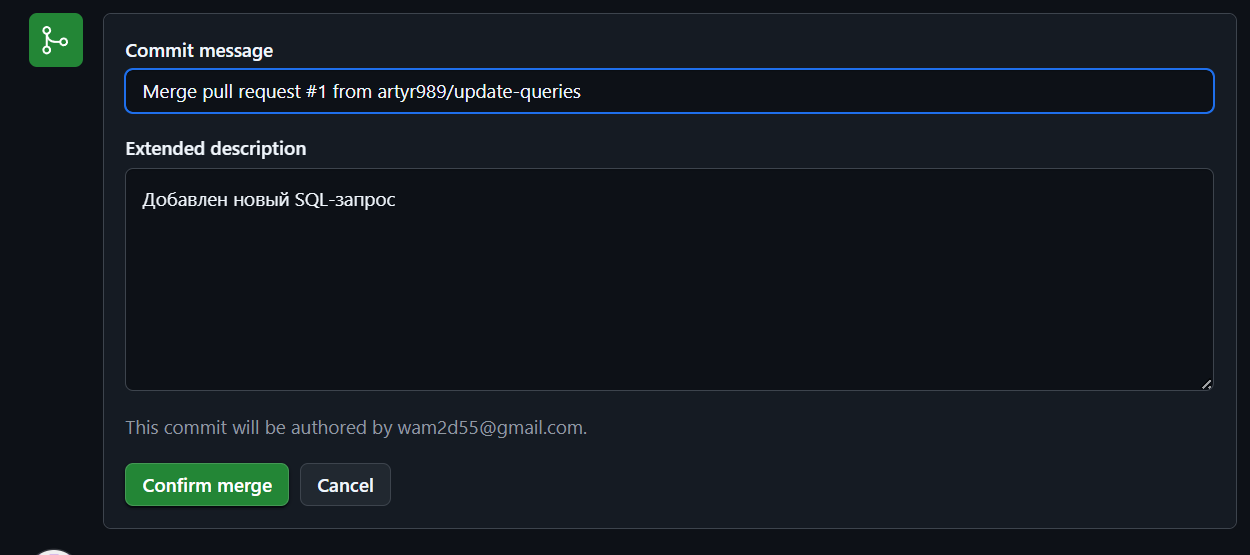


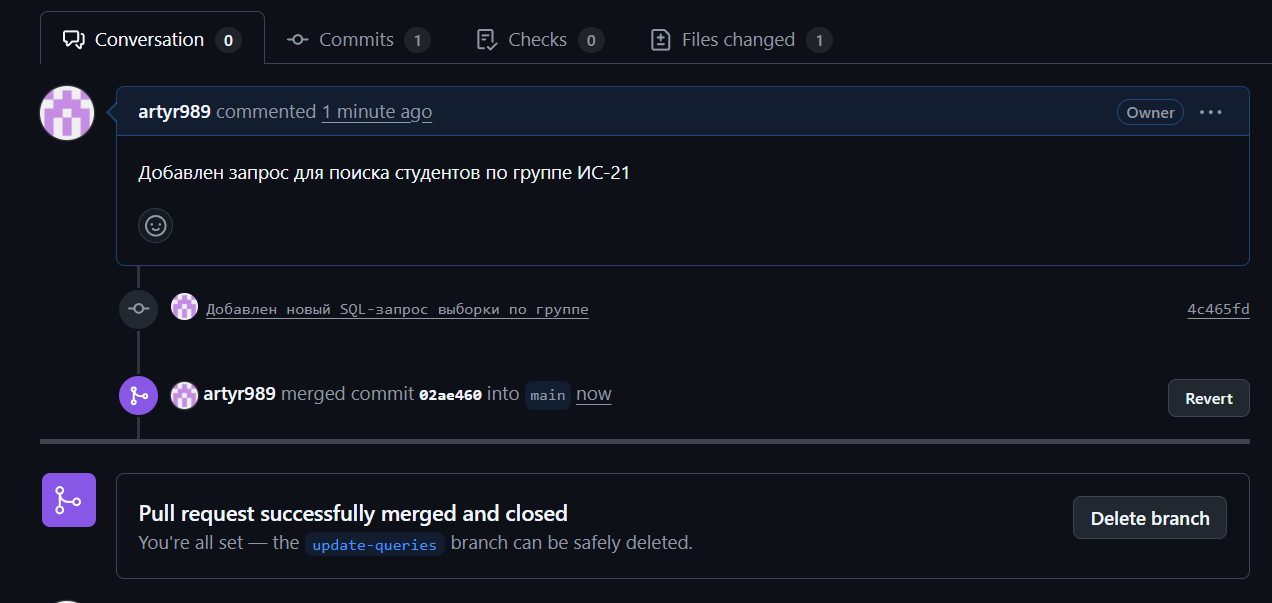
**СОЗДАНИЕ PULL REQUEST НА GITHUB**



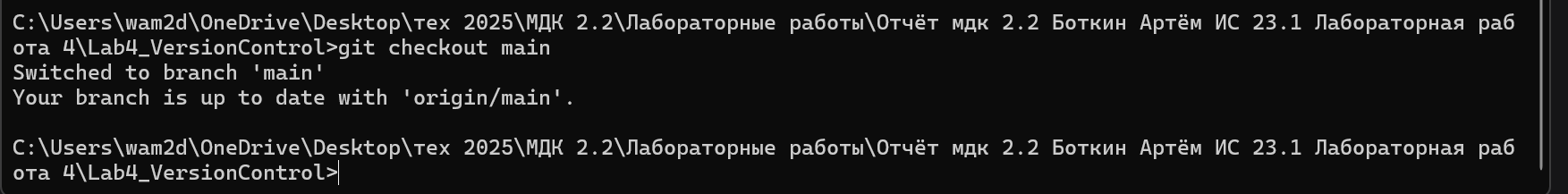


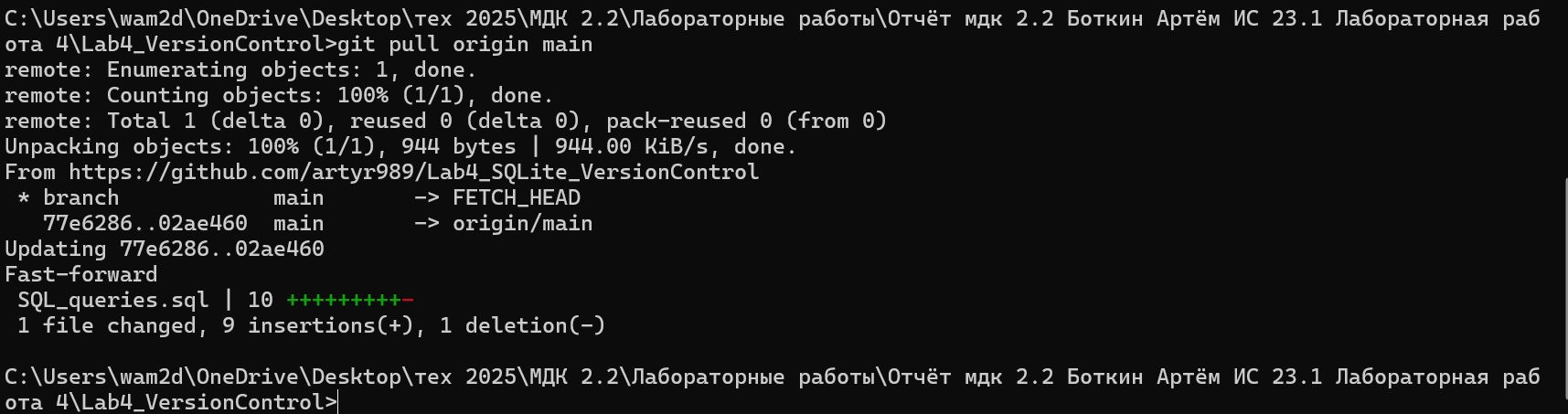






**ПРОВЕРКА**





**Контрольные вопросы**

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

Вкладка "Создание" → "Конструктор запросов" → "Режим SQL". Откроется окно для ручного ввода SQL-запросов.

**2. Какие типы запросов поддерживает Access SQL?**

SELECT (выборка данных)

INSERT (добавление)

UPDATE (изменение)

DELETE (удаление)

CREATE TABLE (создание таблиц)

JOIN (объединение таблиц)

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

Это файл, который указывает Git, какие файлы и папки не нужно отслеживать и добавлять в репозиторий (например, временные файлы, backup-копии).

**4. Какие файлы Access нужно исключать из репозитория?**

.accdb (основная база)

.laccdb (файлы блокировок)

.accde (исполняемые файлы)

.bak (резервные копии)

**5. Что такое ветвление (branch) и зачем оно используется в Git?**

Ветка - это отдельная линия разработки. Позволяет работать над разными функциями одновременно, не мешая основной версии проекта. Например, можно создать ветку для исправления ошибок, а другую - для добавления новой функции.

**6. Как объединить изменения из разных веток?**

Через Pull Request на GitHub или командой git merge. Сначала переключаемся на основную ветку (git checkout main), затем объединяем (git merge имя\_ветки).