МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Институт информационных технологий\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

наименование института (факультета)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_МПО ЭВМ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

наименование кафедры \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Базы данных\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ наименование дисциплины в соответствии с учебным планом

ЛАБОРАТОРНАЯ РАБОТА № 4

УПРАВЛЕНИЕ ТРАНЗАКЦИЯМИ. ЖУРНАЛИЗАЦИЯ ИЗМЕНЕНИЙ БД

Исполнитель

студент \_\_1ПИб-02-1оп-22\_\_

группа

\_\_\_Микуцких Г. А.\_\_\_

Фамилия, имя, отчество

Руководитель \_\_\_Селяничев О.Л.\_\_\_

Ф.И.О. преподавателя

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_2024\_ год

Транзакция – неделимая с точки зрения воздействия на БД последовательность операторов манипулирования данными (чтения, удаления, вставки, модификации), что, либо результаты всех операторов, входящих в транзакцию, отображаются в БД, либо воздействие всех этих операторов полностью отсутствует.

Транзакция – единица активности пользователя. Для многопользовательских систем пока первый пользователь обращается к БД, другому доступ закрыт.

Девиз «Всё или ничего»: при завершении операций оператором «COMMIT» результат фиксируется в ВЗУ, «ROLLBACK» – всё отменяется.

Журнализация – это процесс ведения журнала событий или изменений, происходящих в системе.

С одной БД могут одновременно работать несколько пользователей или прикладных программ. Система должна обеспечивать изолированность пользователей, то есть создание иллюзии работы с БД в одиночку. Тогда транзакция выступает единицей изолированности пользователей.

Бывают ситуации, при которых целостность БД невозможно не нарушить, выполняя только один оператор изменения БД. Поэтому для поддержания целостности допускается ее нарушение внутри транзакции с условием, чтобы к завершению транзакции целостность была соблюдена. Несоблюдение этого условия приводит к тому, что вместо фиксации результатов транзакции происходит её откат (вместо оператора COMMIT выполняется ROLLBACK), и БД остается в таком состоянии, в котором находилась к моменту начала транзакции, то есть в целостном состоянии.

Различают два вида ограничений целостности: немедленно проверяемые и откладываемые.

Немедленно проверяемые ограничения целостности (НПОЦ) – такие ограничения, проверку которых бессмысленно или невозможно откладывать. Пример: возраст сотрудника - 150 лет. Они реализуются отдельным операторами языкового уровня СУБД. При их нарушениях не производится откат транзакции, а лишь отвергается соответствующий оператор.

Откладываемые ограничения целостности (ООЦ) – ограничения на БД, а не на отдельные операции. Эти ограничения проверяются в конце транзакции, и их нарушение вызывает автоматическую замену оператора COMMIT на оператор ROLLBACK. В некоторых системах существует оператор насильственной проверки ограничения целостности внутри транзакции.

Пример БД представлен на рис. 1.

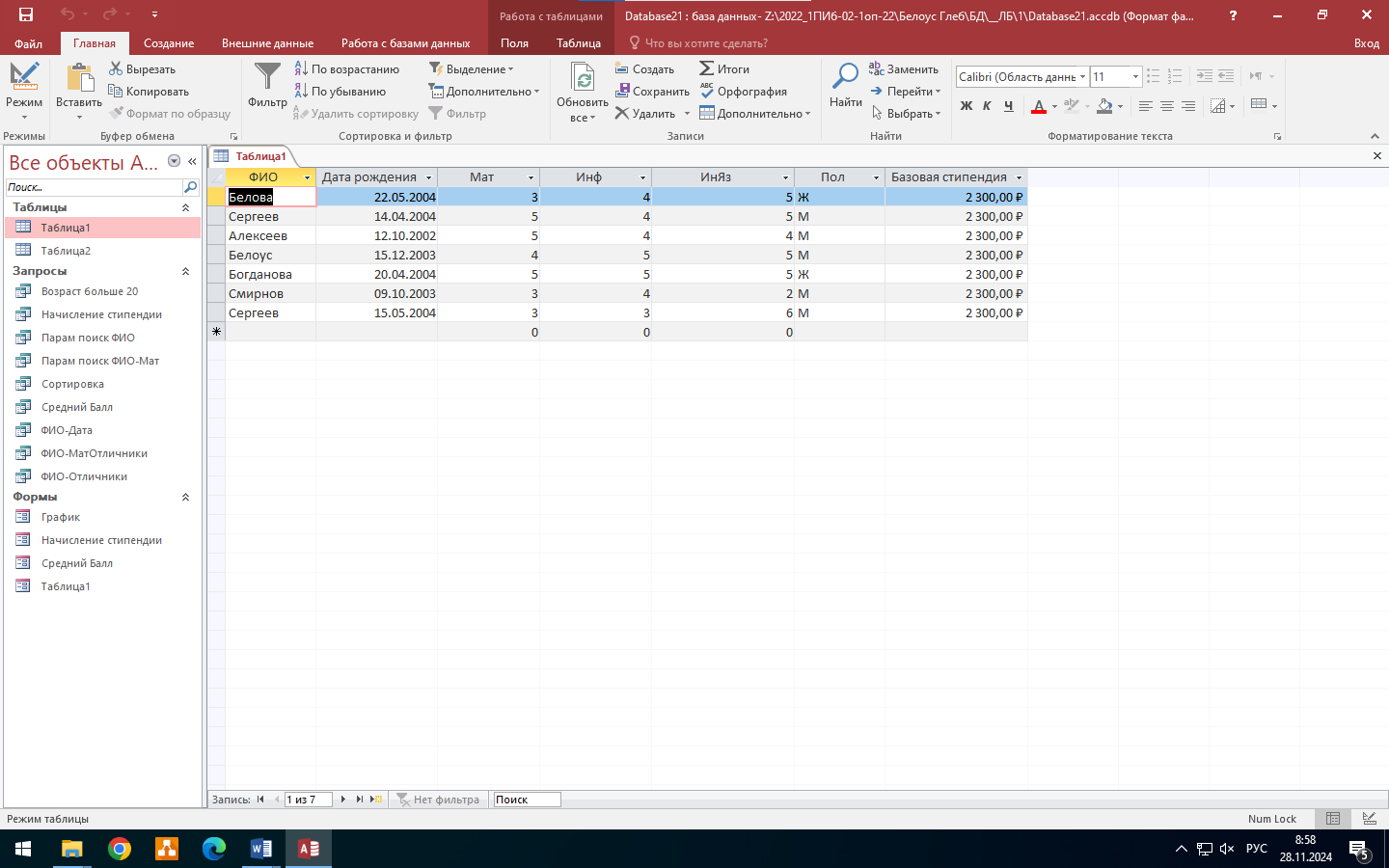


Рис. 1. Вид исходной БД в Access

1. Исследовать работу с одной БД средствами СУБД Access. Изучить, каким образом осуществляется незаметная для пользователей работа для случаев:

- потерянных изменений;

Т1 (первая транзакция) изменяет объект А. До завершения Т1 вторая транзакция Т2 также пытается изменить объект А. Т2 завершается операцией ROLLBACK. Тогда при повторном чтении объекта Т1 не видит изменений объекта. Это противоречит изолированности пользователей.

Проблема: ситуация потерянных изменений.

Решение: до завершения Т1 транзакции, никакая другая транзакция не может изменить объект.

Профилактика: отсутствие потерянных изменений – минимальное требование к СУБД при синхронизации параллельно выполняемых транзакций.

Любые изменения в БД Access (как и свойств ячеек таблицы) считаются транзакцией. Если БД открыта у двух пользователей и T1 с T2 идут параллельно, то будет потеря одной из транзакций при попытке сохранения.

Если во время изменений Т1 (первой транзакцией) Т2 произведёт изменения и сохранит их, то при сохранении Т1 будет выведено следующее окно (рис. 2).

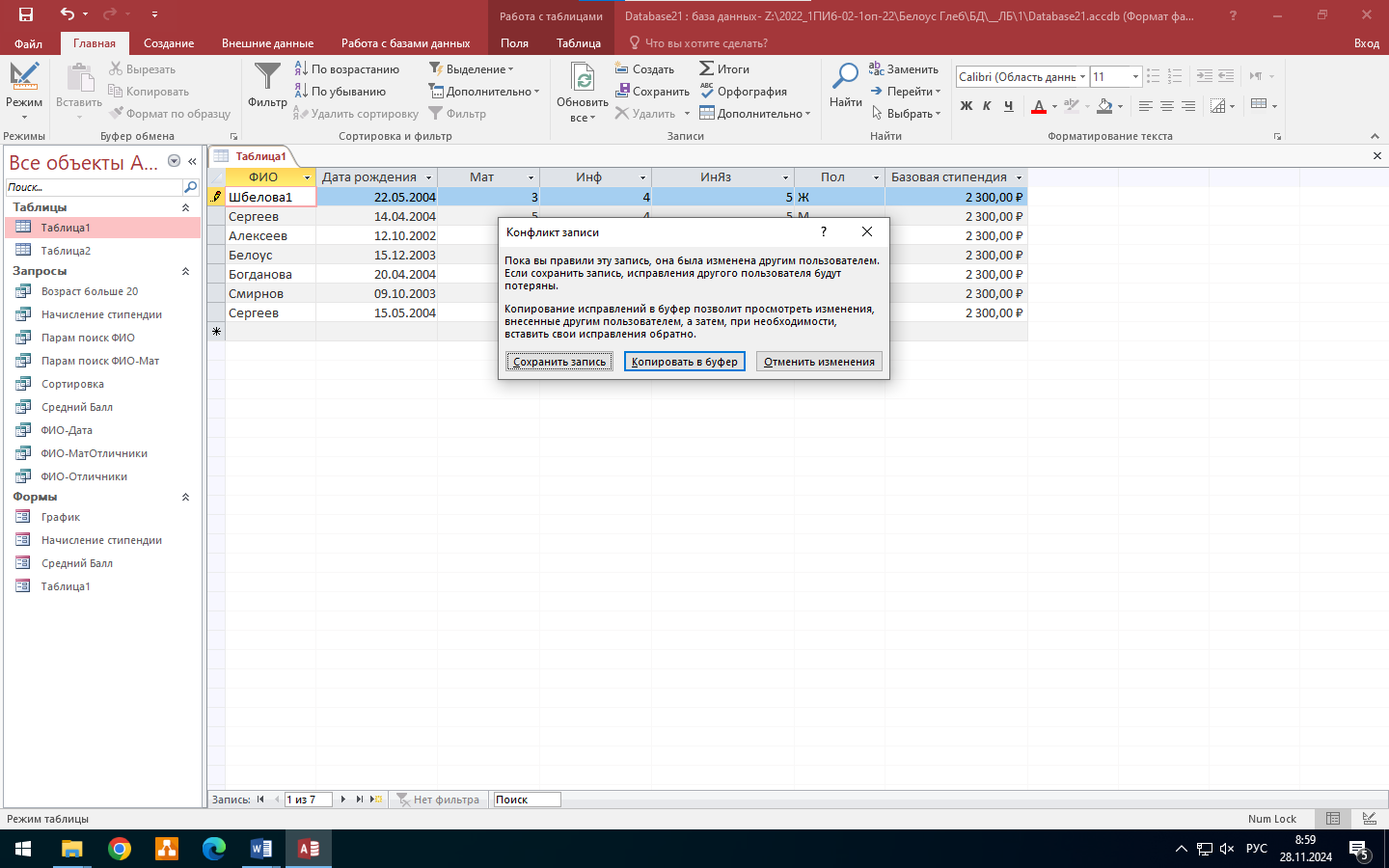


Рис. 2. Реакция Access на конфликт записи для Т1

Если Т1 завершит изменения до сохранения Т2 (Т2 начата позже), то Access выведет следующее диалоговое окно для Т2 (рис. 3).

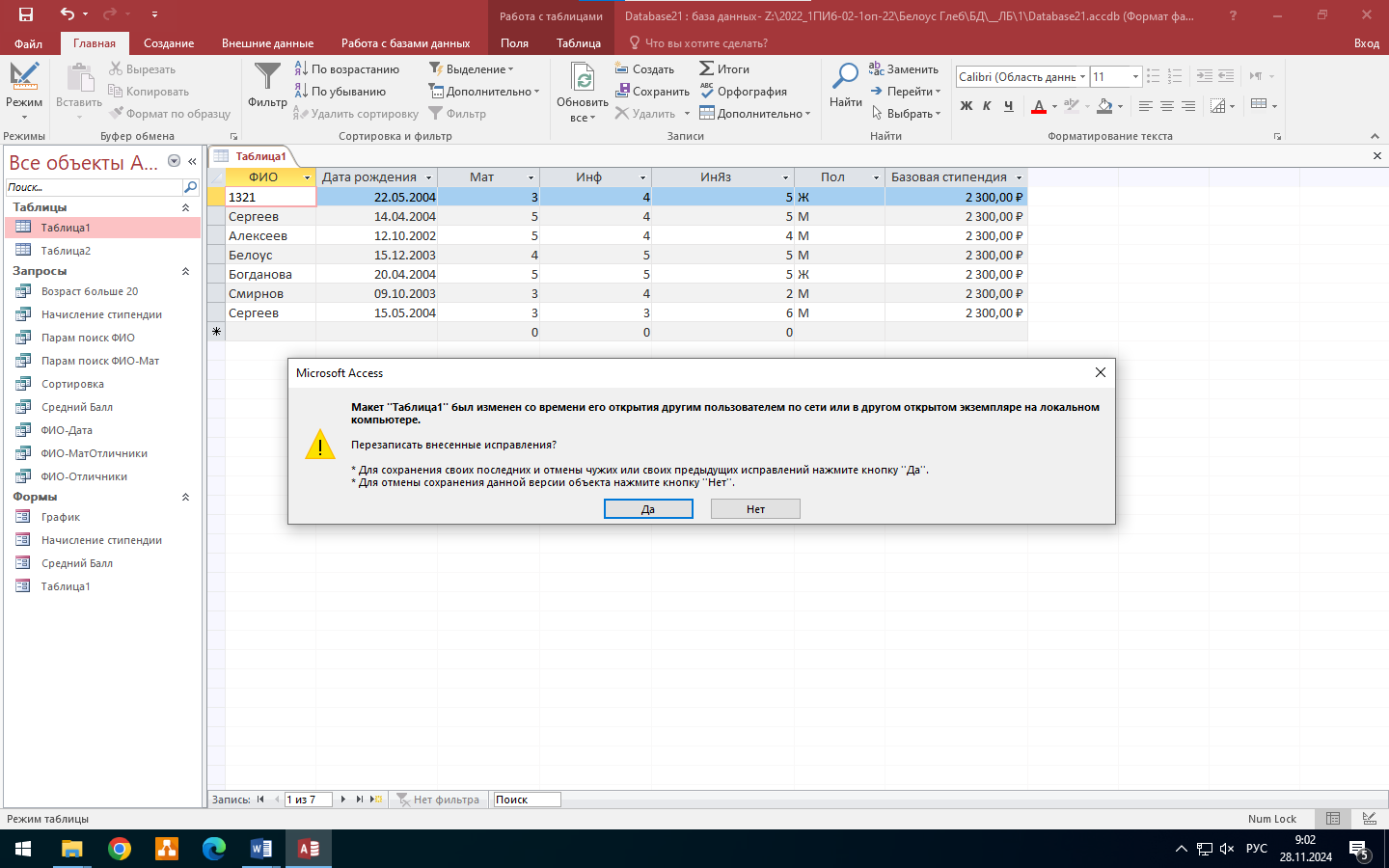


Рис. 3. Предупреждение Access о потере изменений

- чтение «грязных» данных;

Т1 изменяет объект, параллельно с этим Т2 читает объект A. Так как операция изменения не завершена, Т2 видит несогласованные данные.

Проблема: несоответствие требованию изолированности пользователей, где каждый пользователь начинает свою транзакцию при согласованном состоянии базы данных и вправе видеть согласованные данные.

Решение: до согласования транзакции объекта Т1, другая транзакция не должна читать объект, где минимальным требованием является блокировка чтения объекта А до завершения операции его изменения в Т1.

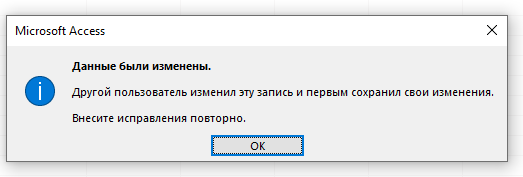


Рис. 14. Реакция Access на чтение «грязных» данных

- неповторяющихся чтений;

Т1 читает объект БД А. До завершения первой транзакции Т2 изменяет объект А и успешно завершается COMMIT. Т1 повторно читает объект А и видит его изменённое состояние.

Решение: до завершения первой транзакции никакая другая транзакция не должна изменять объект (рис. 5).

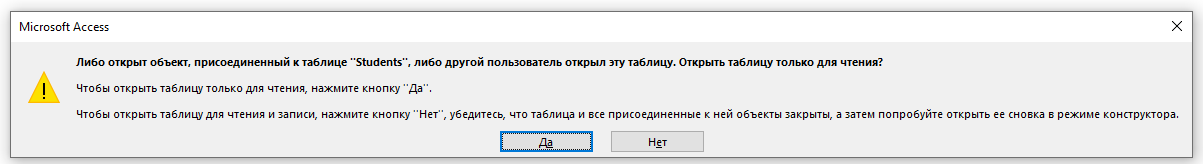


Рис. 5. Реакция Access на чтение БД, пока не завершена T1

Т2 изменяет запись в БД Access во время чтения Т1 и нажимает «Enter», чтобы подтвердить изменение. Одиночное нажатие не вызовет COMMIT, но если поле, в котором происходит изменение, является крайним справа, то нажатие «Enter» приведёт к переходу на новую запись, что вызовет COMMIT.

Следующие манипуляции также вызовут COMMIT:

1. Нажатие «F5» (переход к первой записи).
2. Клавиши «Page Up» и «Page Down» (переход к предыдущей и следующей записям соответственно).
3. Нажатие «Сохранить» в разделе «Файл».
4. Нажатие иконки «Сохранить» в разделе «Главное».
5. Нажатие значка «Сохранить» в левом верхнем углу программы.
6. Нажатие «Tab» (переход к следующему полю записи, для перехода к предыдущему используется комбинация «Shift» + «Tab»).
7. Нажатие комбинации «Ctrl» + «S».

Изучить, каким образом осуществляется незаметная для пользователей работа для случая:

- кортежей-фантомов.

Т1 выполняет оператор А выборки кортежа отношения R («Relationship») c условием выборки S («Share»). До завершения первой транзакции Т2 вставляет в R новый кортеж, удовлетворяющий S, и совершает COMMIT. Т1 повторно также выполняет этот оператор и в результате видит кортеж, которого не было.

Проблема: противоречие идеи изолированности.

Решение: требуется более высокий уровень синхронизации транзакции. Пример появления кортежа-фантома представлен на рис. 6-8.

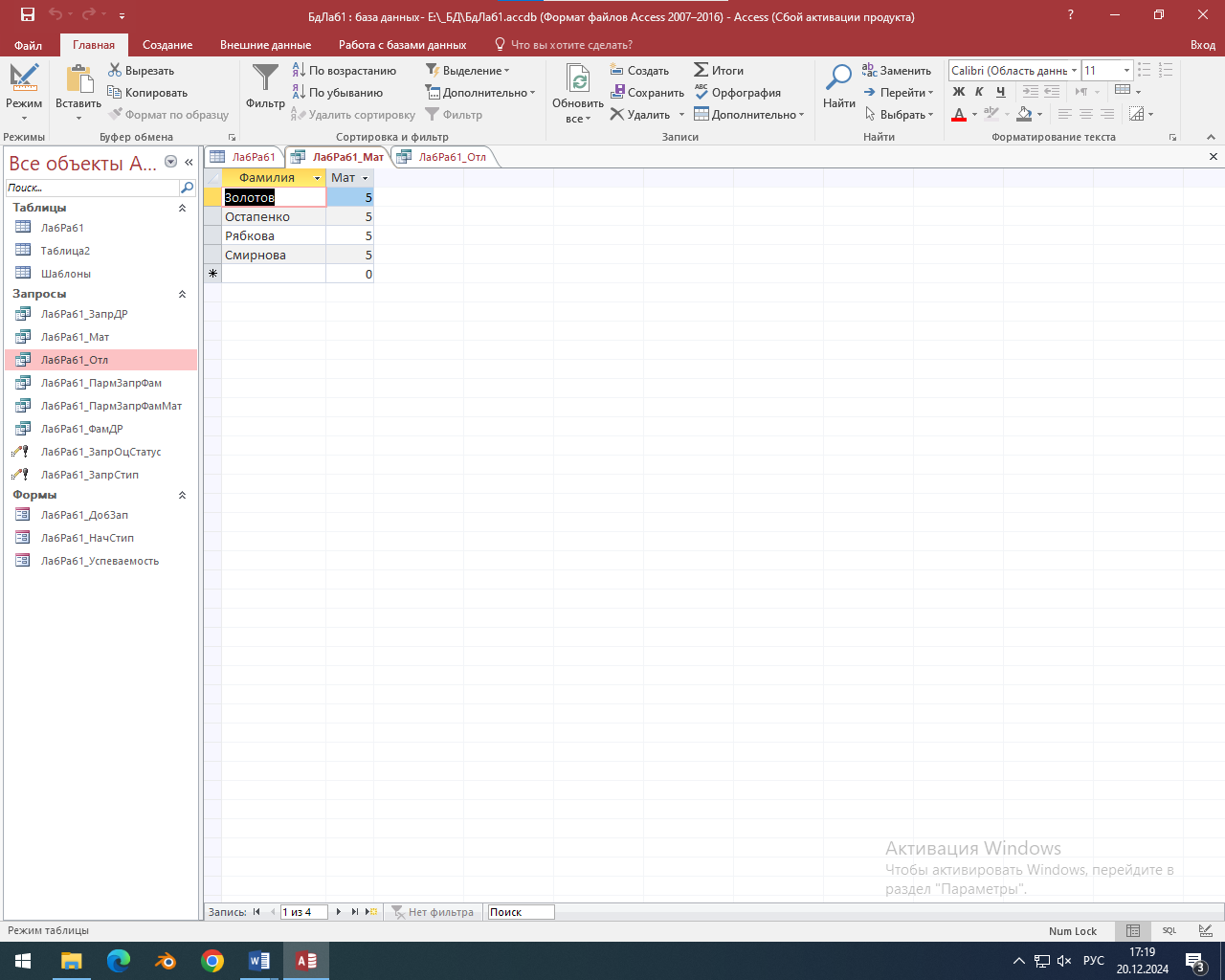


Рис. 6. Результат запроса до ввода новой записи

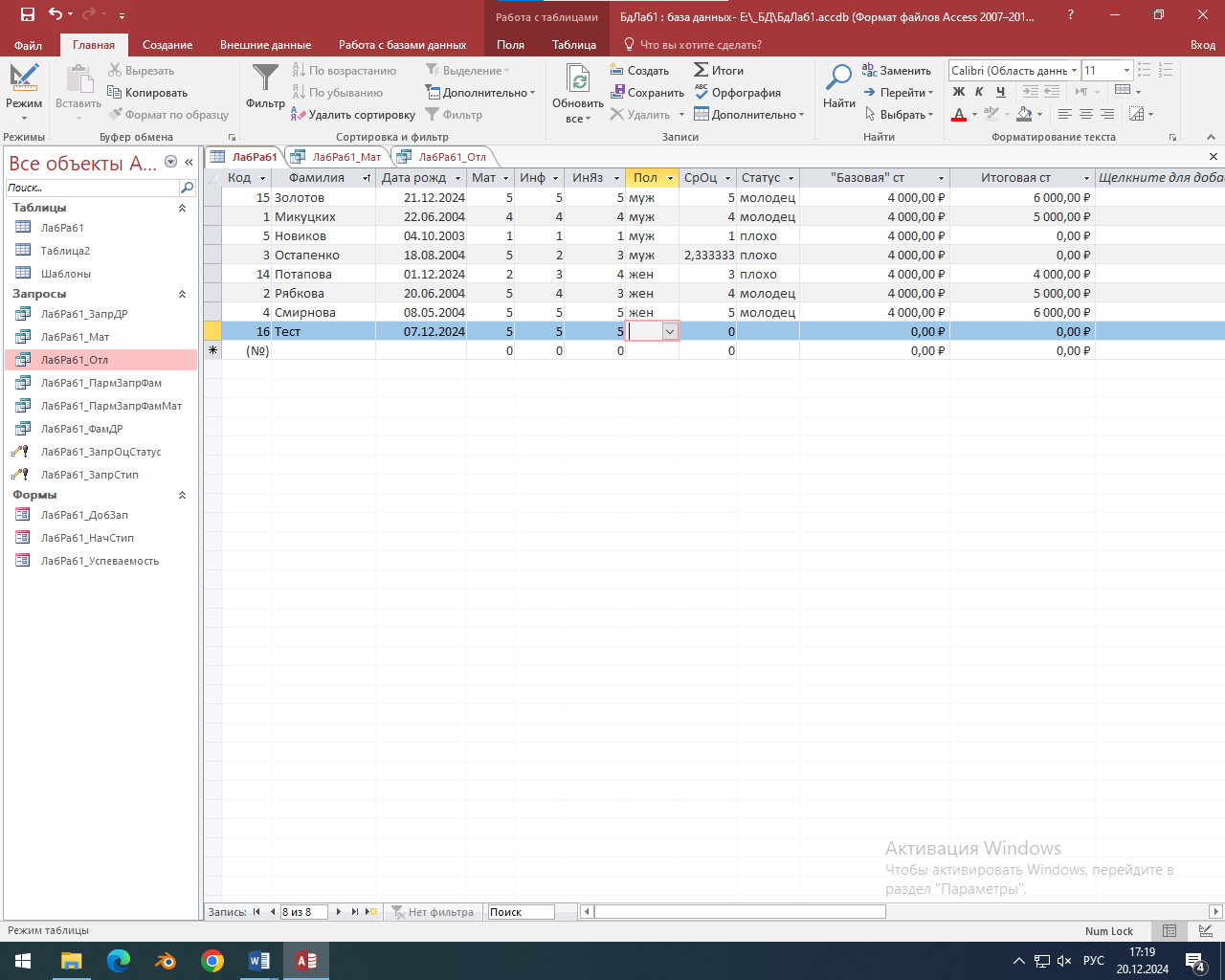


Рис. 7. Введена новая запись «Тест», удовлетворяющая запросу

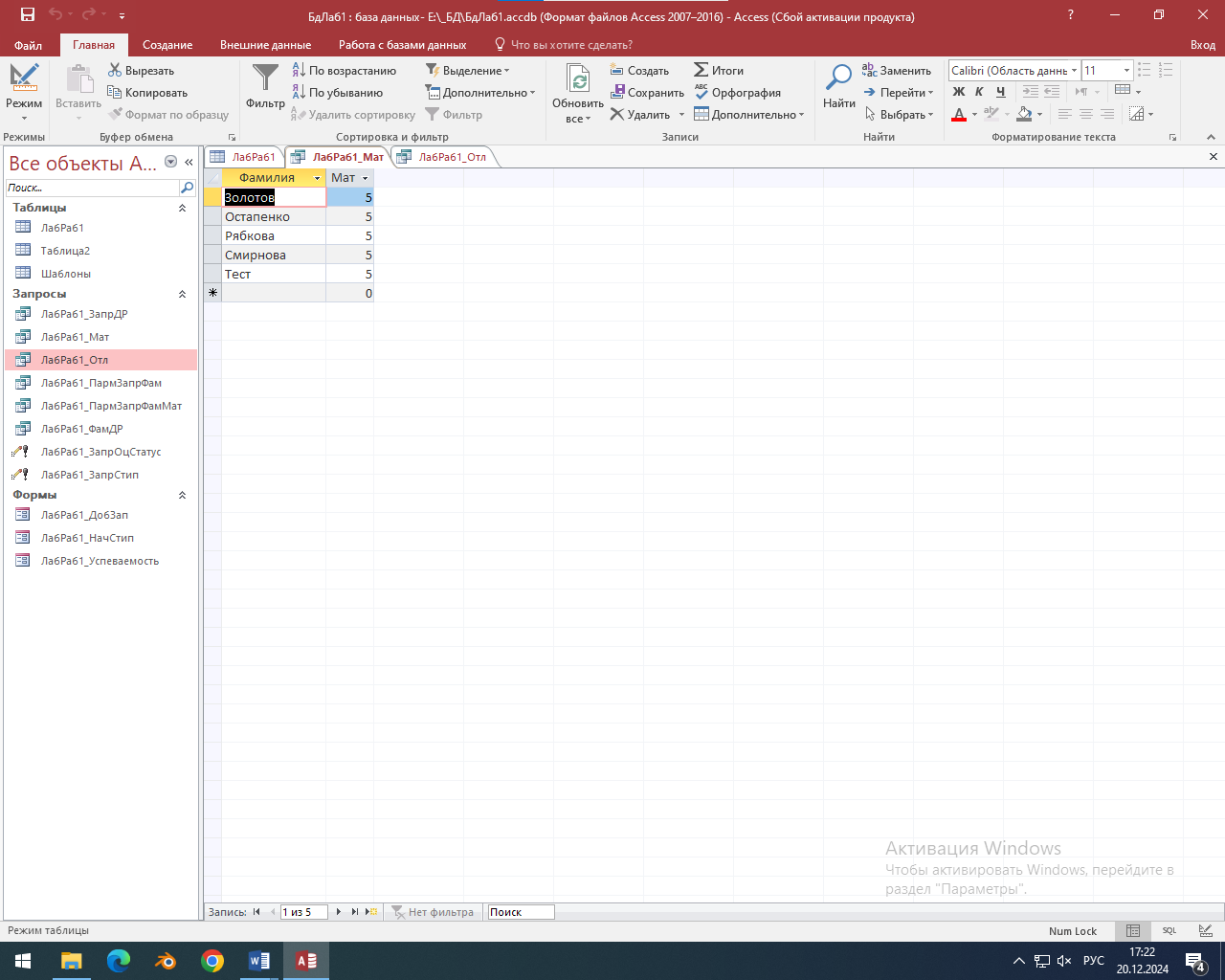


Рис. 8. Результат запроса после повторного выполнения

2. На примере эксплуатации конкретной БД рассмотреть поведение СУБД, направленное на поддержание целостности БД, для случаев:

- отката транзакции;

Access обеспечивает механизм «всё или ничего», автоматически откатывая изменения в случае возникновения ошибок, тем самым поддерживая целостность данных. Этот автоматический откат является ключевым механизмом поддержания целостности в Access. За счёт этого механизма базу данных невозможно привести в несогласованное состояние из-за одиночных сбоев запросов. Откат можно сделать оператором ROLLBACK, который вручную вызывается следующим образом:

1. нажатие клавиши «Esc»;
2. комбинация клавиш «Crtl» + «Z»;
3. нажатие стрелки «Отменить ввод» в левом верхнем углу экрана;
4. автоматически, если отсутствует монопольный доступ к БД.

Пример отката транзакций представлен на рис. 9-13.

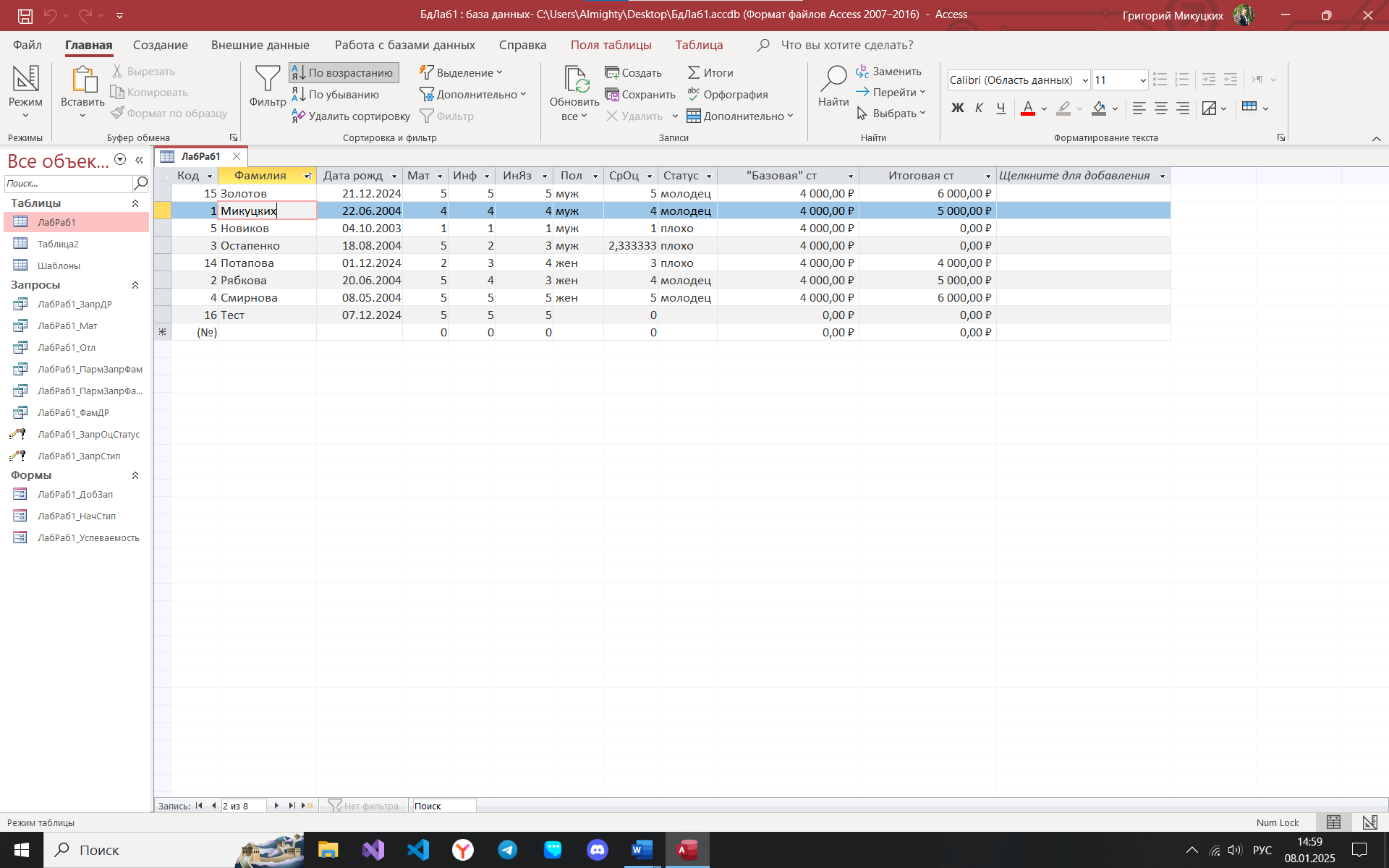


Рис. 9. Вид БД до изменений

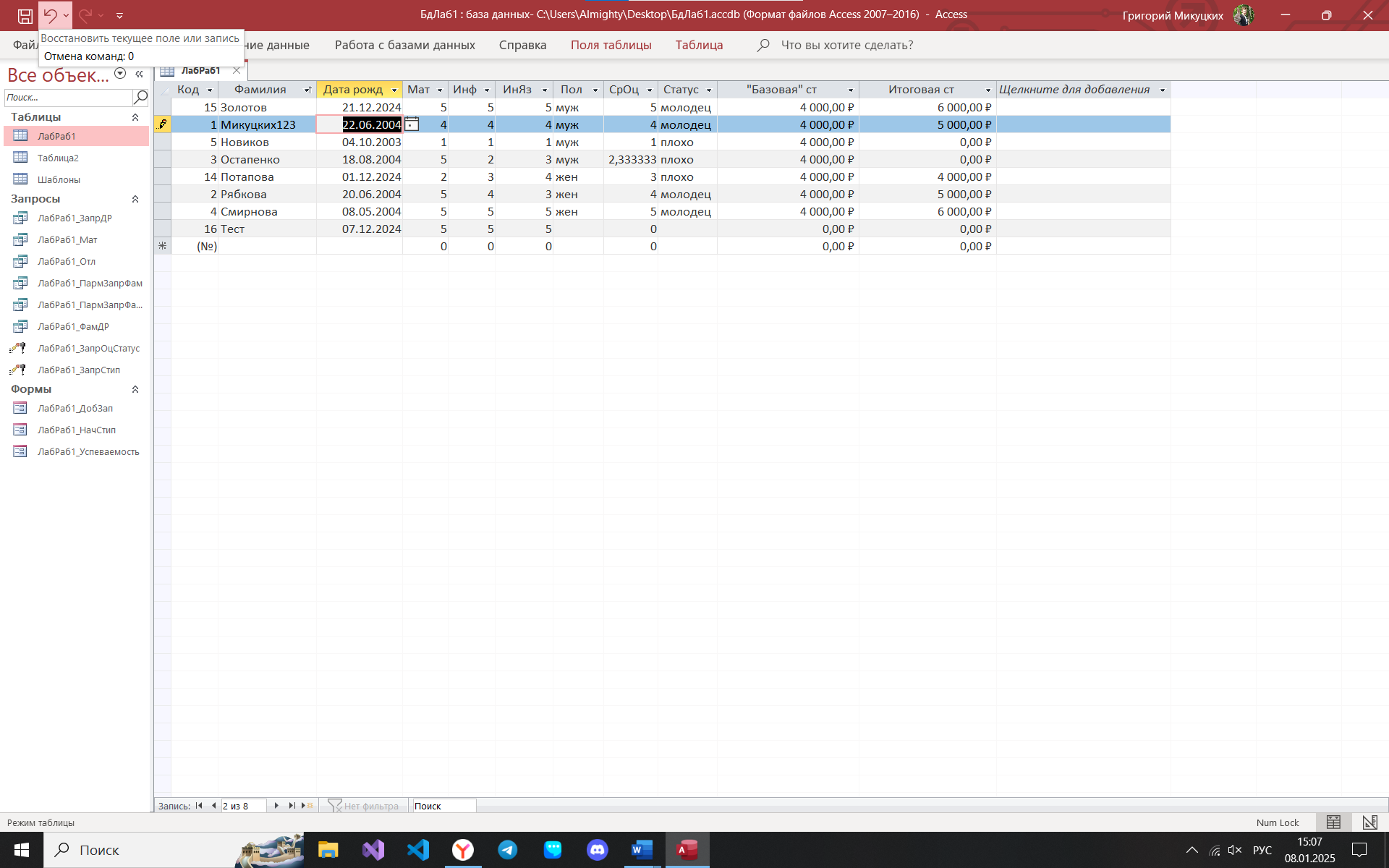


Рис. 10. Кнопка «Отменить» для отката транзакции изменения

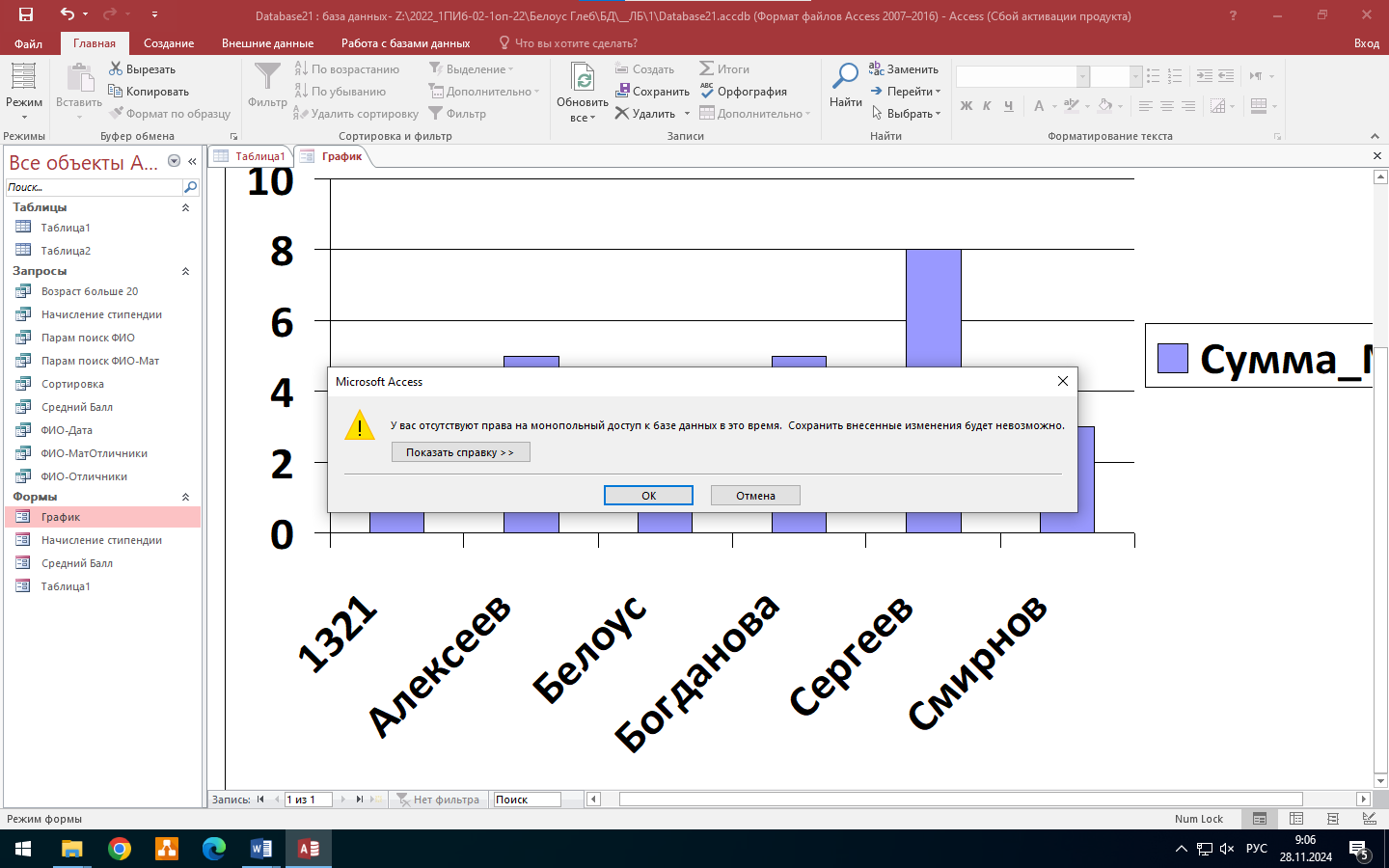


Рис. 12. Пример отката транзакции 2

(открытие «Конструктора» формы пока изменяли БД)

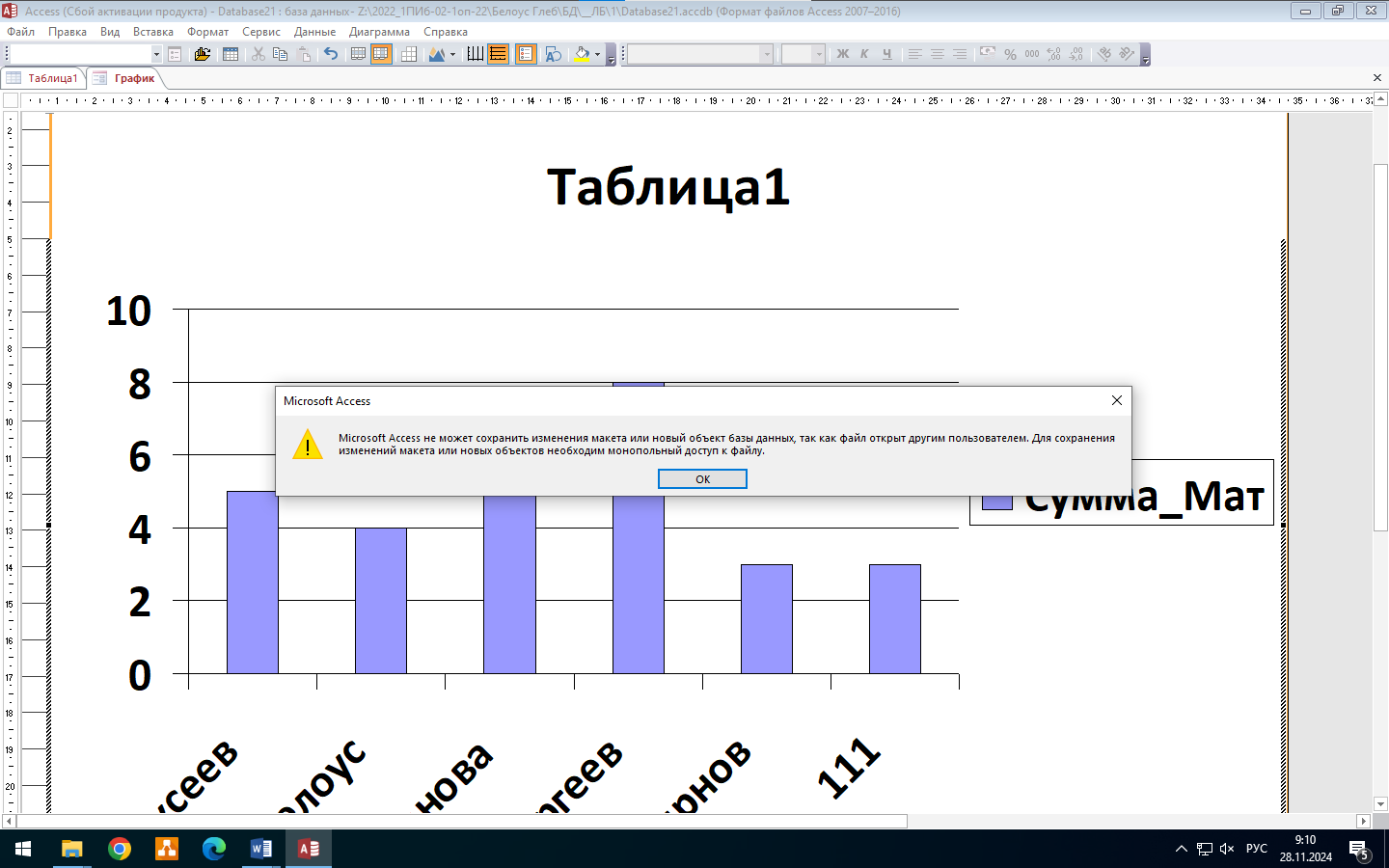


Рис. 13. Пример отката транзакции 3

(одновременная работа в «Конструкторе» формы)

- мягкого сбоя;

В Access «мягкий сбой» обычно означает ситуацию, когда БД повреждена, но не настолько сильно, чтобы стать полностью недоступной. Это может произойти из-за внезапного отключения питания, сбоя программного обеспечения, сбоя процессора или ошибок в работе самой базы данных или принудительного закрытия через «Диспетчер задач». Поведение Access при мягком сбое направлено на минимизацию потери данных и восстановление целостности, то есть произойдёт потеря содержимого ОЗУ (потеря текущей транзакции).

- жесткого сбоя (без его реализации).

В контексте Access «жесткий сбой» обычно означает такое повреждение файла БД, которое делает его полностью недоступным. Это может быть вызвано:

1. Серьезным повреждением файловой системы.
2. Физическим повреждением носителя.
3. Вирусной атакой.
4. Неправильным завершением работы операционной системы (резкое отключение питания или аварийное завершение работы ОС).

Выполняется просмотр журнала транзакций для выявления всех транзакций, которые закончились успешно до наступления сбоя. Транзакции, которые закончились не успешно, не сохранились. По журналу транзакций повторяются все успешно законченные транзакции.

В отличие от мягких сбоев, при жестком сбое встроенные механизмы восстановления Access часто оказываются недостаточными (запись всех изменений данных, включая вставку, обновление и удаление записей, возможность установки точек сохранения внутри транзакции, откат транзакций). Поведение СУБД в этом случае зависит от степени повреждения:

1. Полная недоступность: Access может выдать сообщение об ошибке, например «Невозможно открыть базу данных». В этом случае файл базы данных поврежден настолько, что Access не может его прочитать или восстановить.
2. Частичная доступность с ошибками: в некоторых случаях Access может открыть базу данных, но при этом будут возникать ошибки, например, невозможность открыть определенные таблицы или запросы. Это указывает на частичное повреждение данных или структуры базы данных.

Восстановление после жесткого сбоя:

1. Восстановление из резервной копии. Если есть свежая резервная копия, восстановление будет наиболее полным.
2. Использование утилит восстановления данных. Эти утилиты могут попытаться восстановить данные даже из сильно поврежденных файлов, но гарантий нет, и восстановленные данные могут быть неполными или поврежденными.