**ЛАБОРАТОРНАЯ РАБОТА № 22**

**«Разработка триггеров для таблиц базы данных – поддержание ссылочной целостности»**

Реализовать триггеры для поддержания ссылочной целостности базы данных. В отчете должны быть представлены SQL-запросы для удаления описания некоторых связей между таблицами и создания триггеров для замены этих описаний. Должны быть разработаны триггеры следующих видов:

1)      для каскадного удаления записей дочерней таблицы при удалении записи родительской таблицы;

2)      запрета на удаление записи родительской таблицы при наличии связанных записей в дочерней таблице;

3)      контроля наличия записи родительской таблицы при занесении записи дочерней таблицы;

4)      контроля наличия записи родительской таблицы при изменении кода этой записи в дочерней таблице.

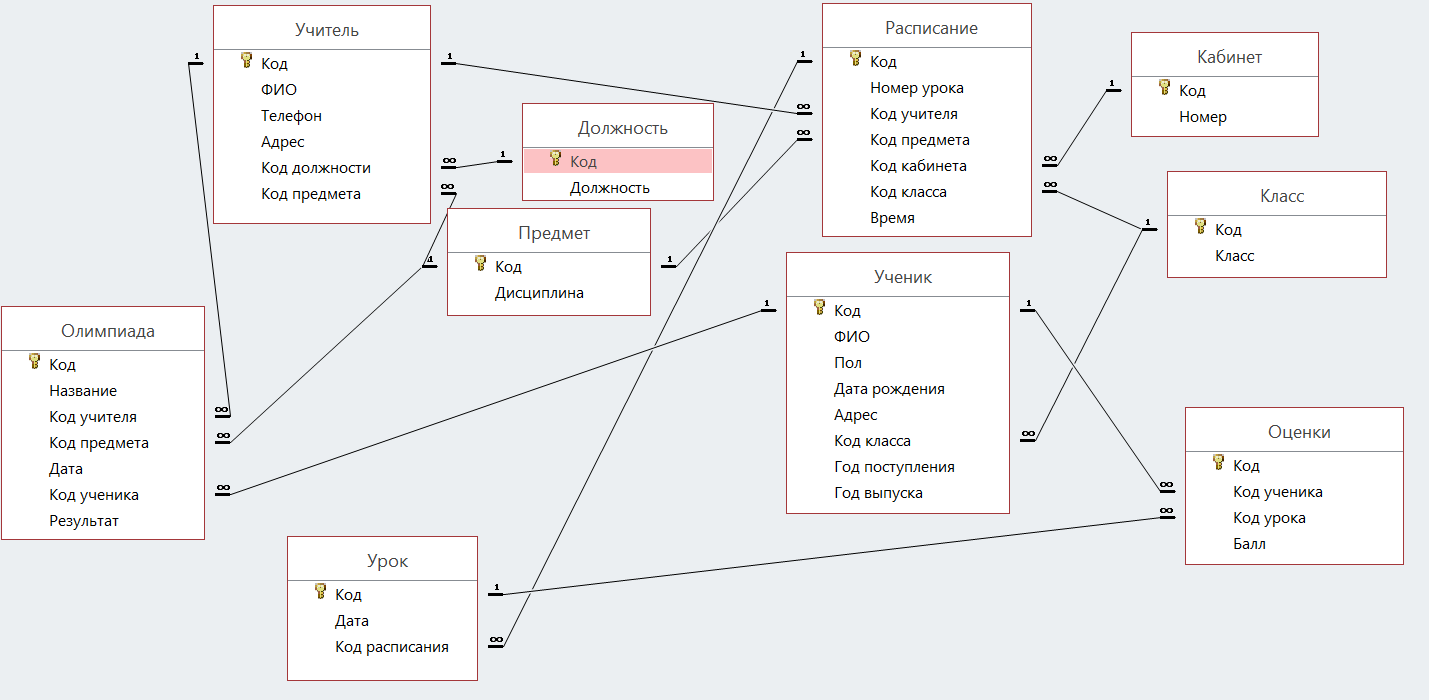
В отчете также должны быть представлены скриншоты, демонстрирующие действие триггеров при работе с базой данных на сервере из MS Access:

1)      для каскадного удаления записей: содержимое дочерней таблицы до удаления записи родительской таблицы, удаление записи родительской таблицы,  содержимое дочерней таблицы после удаления записи родительской таблицы;

2)      для запрета на удаление: сообщение об ошибке при попытке удаления записи;

3)      для контроля наличия записи: соответствующие сообщения об ошибке (может выполняться в режиме отображения дочерней таблицы, когда код записи родительской таблицы вводится «руками»).

**База данных:**

****

Sql-запрос:

Alter Table Оценки

Drop FK\_Оценки\_Урок

Go

Alter Table Ученик

Drop FK\_Ученик\_Класс

Go

Alter Table Расписание

Drop FK\_Расписание\_Класс

Go

Alter Table Оценки

Drop FK\_Оценки\_Ученика

Go

-- 1. Для каскадного удаления записей дочерней таблицы при удалении записи родительской таблицы.

Create trigger trgУрок\_delete on Урок

After delete

As

Delete from Оценки

Where [Код урока] in (Select Код From deleted)

Go

-- 2. Запрета на удаление записи родительской таблицы при наличии связанных записей в дочерней таблице.

Create trigger trgКласс\_delete on Класс

After delete

As

If exists(Select \* from Ученик Where [Код класса] in (Select Код from deleted))

Begin

Raiserror('Нельзя удалить, так как имеются ученики.', 16, 2)

Rollback tran

End

Go

-- 3. Контроля наличия записи родительской таблицы при занесении записи дочерней таблицы.

Create trigger trgЗанесение\_Оценок on Оценки

After insert

As

If exists(Select \* from inserted where [Код ученика] not in (Select Код From Ученик))

Begin

Raiserror('Неверный код ученика.', 16, 2)

Rollback tran

End

Go

-- 4. Контроля наличия записи родительской таблицы при изменении кода этой записи в дочерней таблице.

Create trigger trgИзменение\_Оценок on Оценки

After update

As

If update([Код ученика])

If exists(Select \* from inserted where [Код ученика] not in (Select Код from Ученик))

Begin

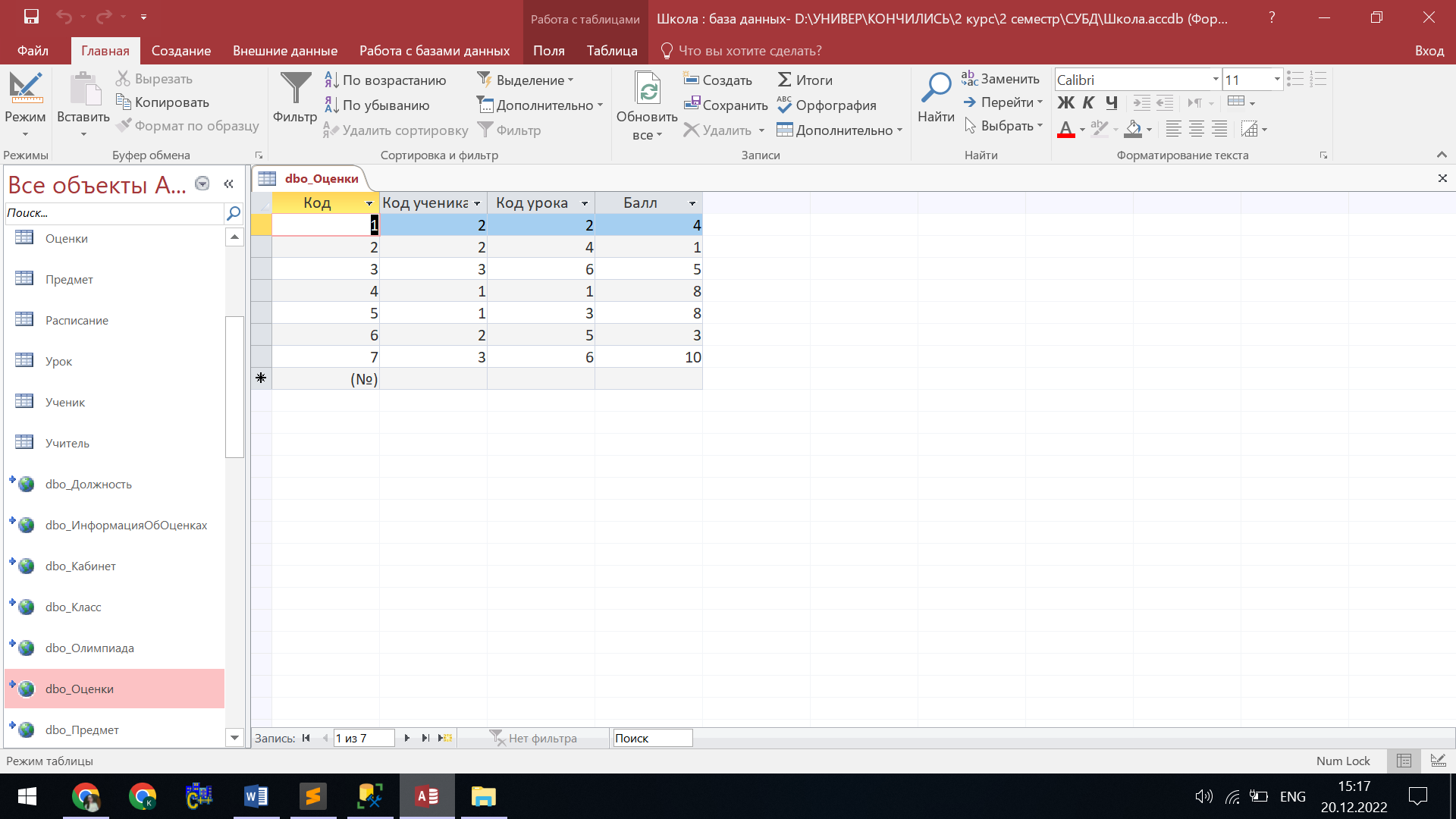
Raiserror('Неверный код ученика.', 16, 2)

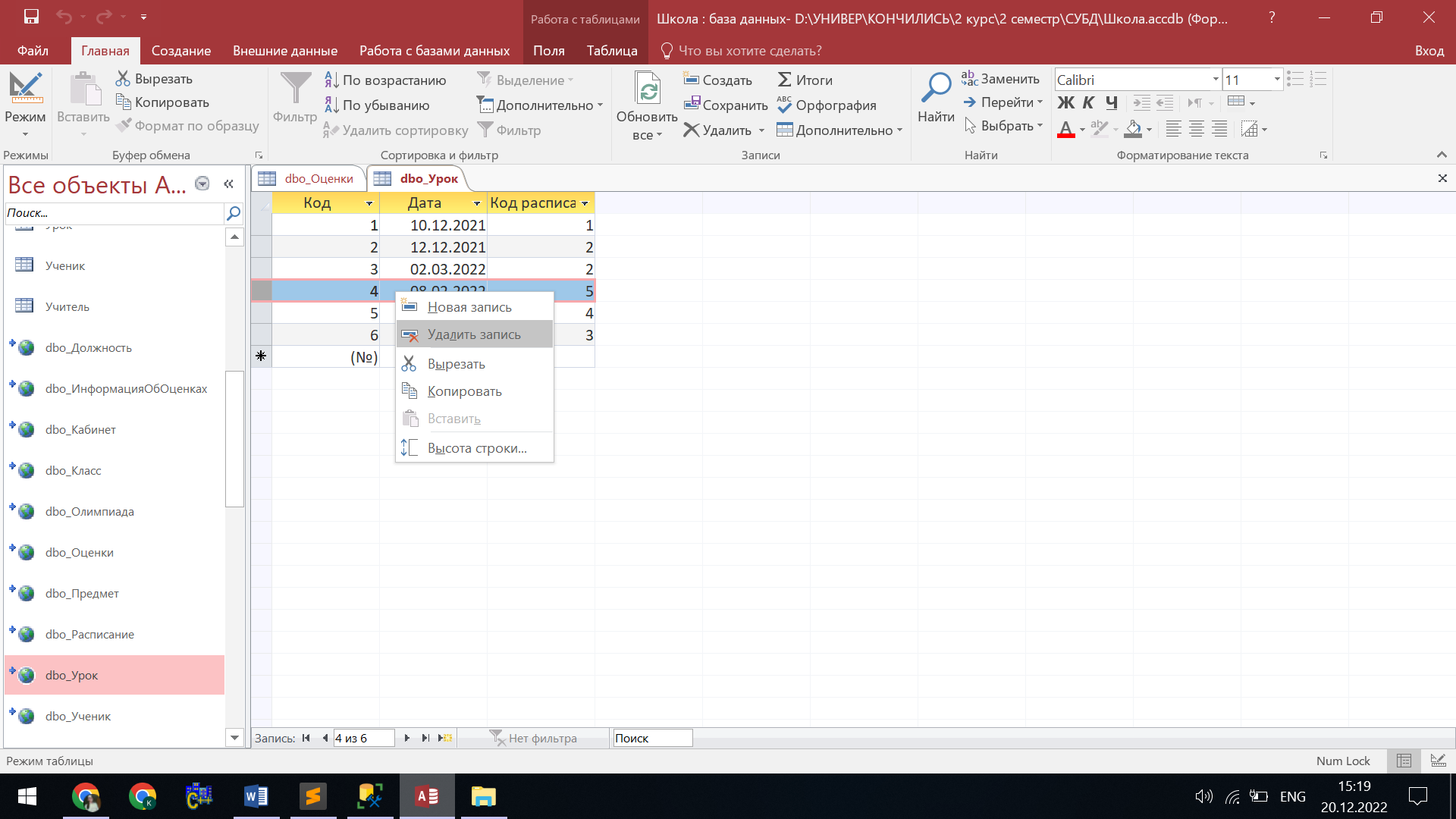
Rollback tran

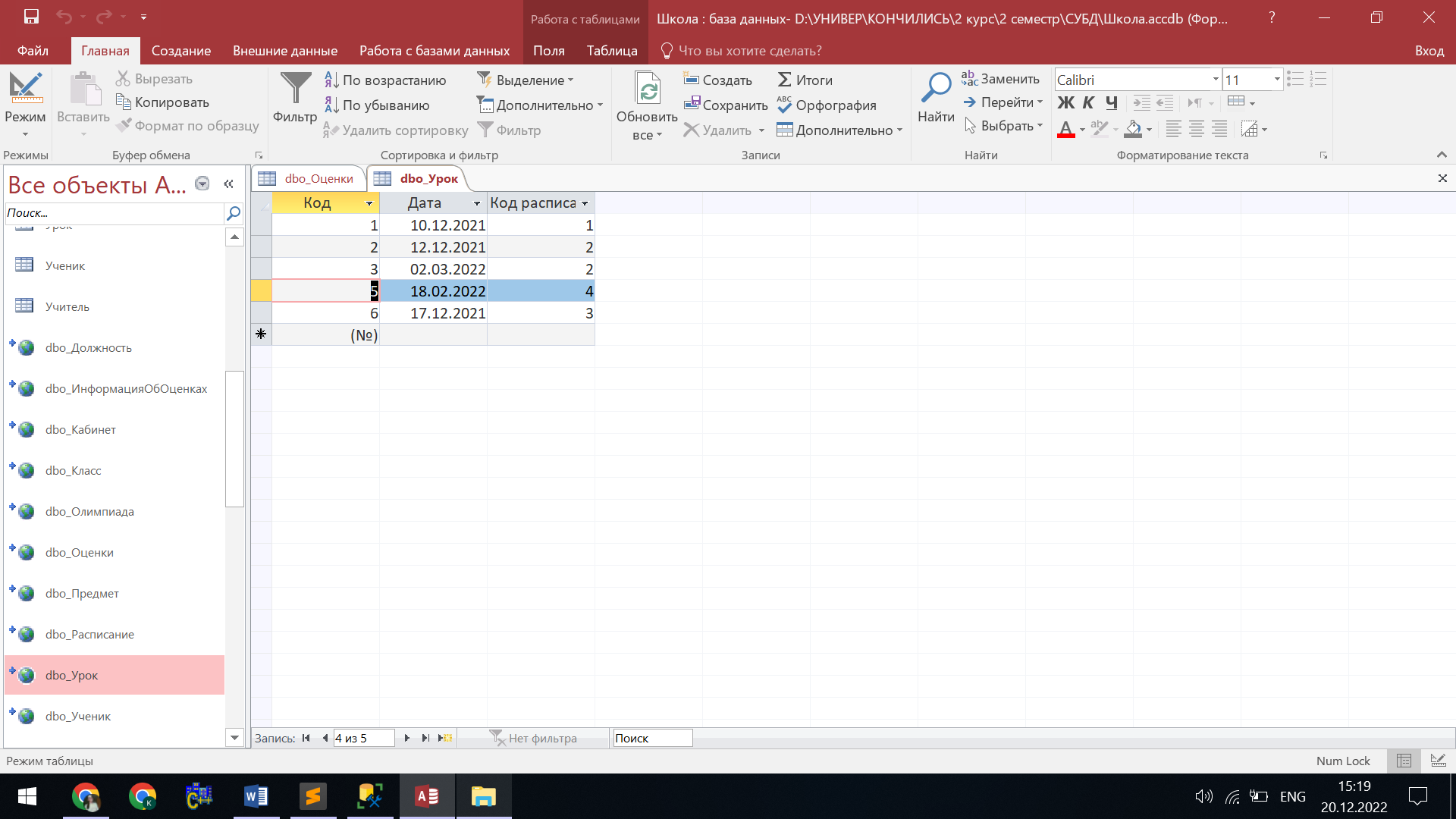
End

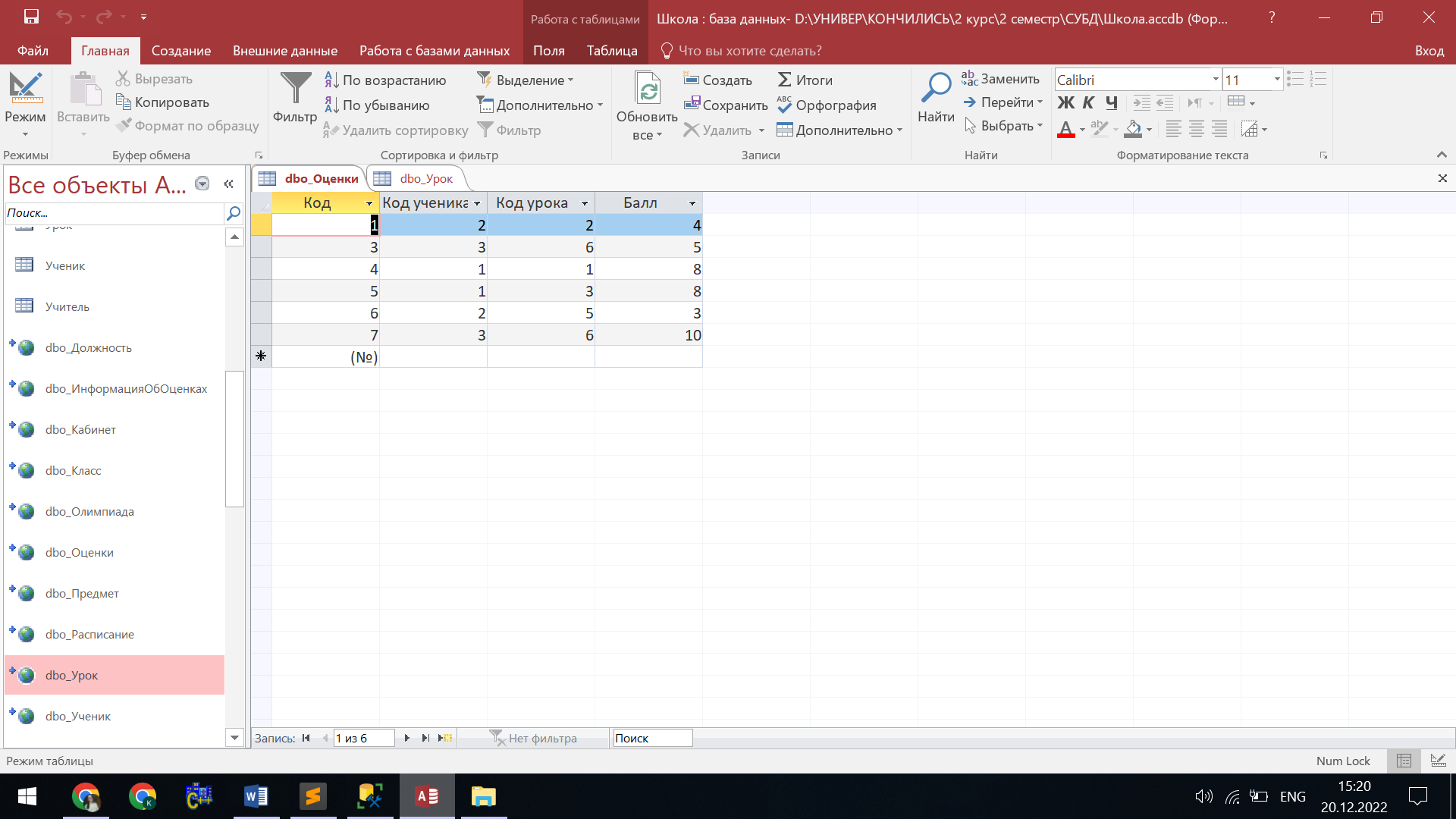
Go

1. Для каскадного удаления записей: содержимое дочерней таблицы до удаления записи родительской таблицы, удаление записи родительской таблицы,  содержимое дочерней таблицы после удаления записи родительской таблицы:

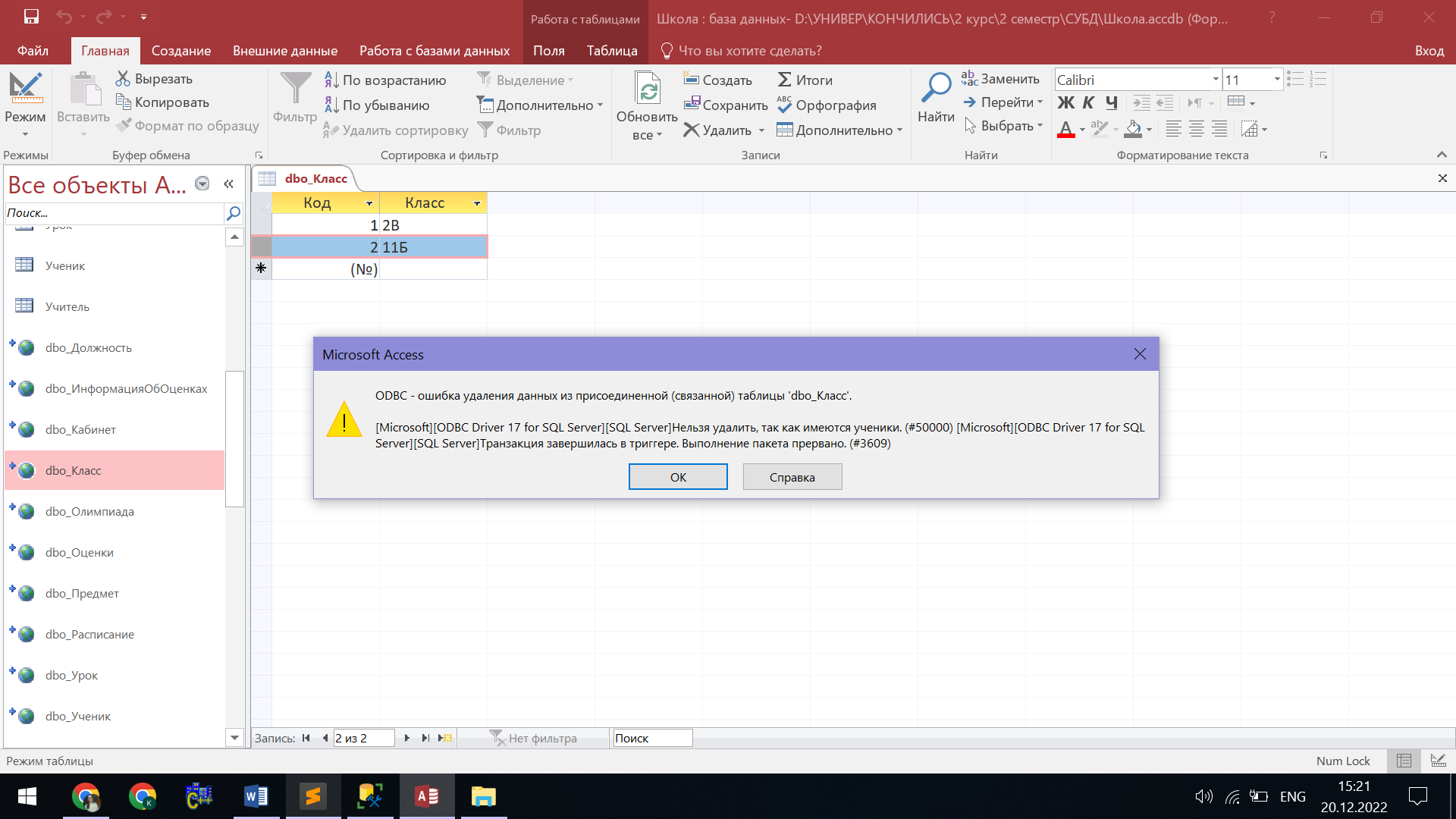








1. Для запрета на удаление: сообщение об ошибке при попытке удаления записи:



1. Для контроля наличия записи: соответствующие сообщения об ошибке (может выполняться в режиме отображения дочерней таблицы, когда код записи родительской таблицы вводится «руками»):

