

Лабораторная работа №21

Команда «Triggers»

Шеховцов Виктор ЗМО

Условие БД:

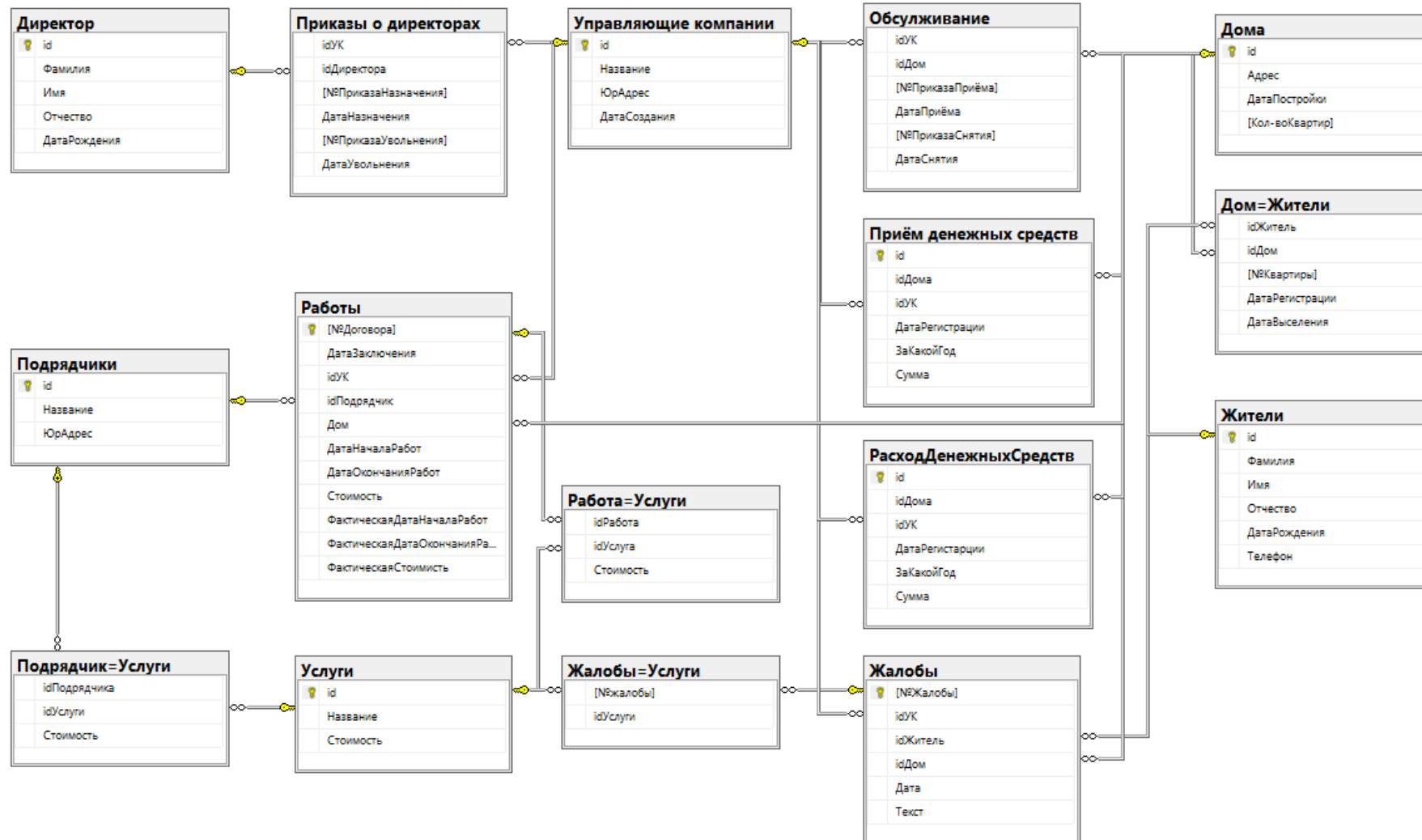
Билет № 8

по дисциплине «Базы данных»

1. Создать базу данных (таблицы базы данных, связи между таблицами) следующей предметной области:

- Название Управляющей компании
- Юридический адрес Управляющей компании;
- Дата создания Управляющей компании
- Фамилия, имя, отчество директора Управляющей компании
- Дата назначения директора на должность
- Дата снятия директора с должности
- Адрес дома, который обслуживает данная компания
- Дата приема на обслуживание
- Дата снятия с обслуживания
- Количество квартир в конкретном доме, обслуживаемом конкретной Управляющей компанией
- Сумма денежных средств, поступившая от жильцов определённого дома, обслуживаемого определённой УК за определённый год
- Сумма денежных средств, потраченных УК для обслуживания определённого дома за определённый год
- Жалоба (дата и адрес дома), поступившая на деятельность определённой УК

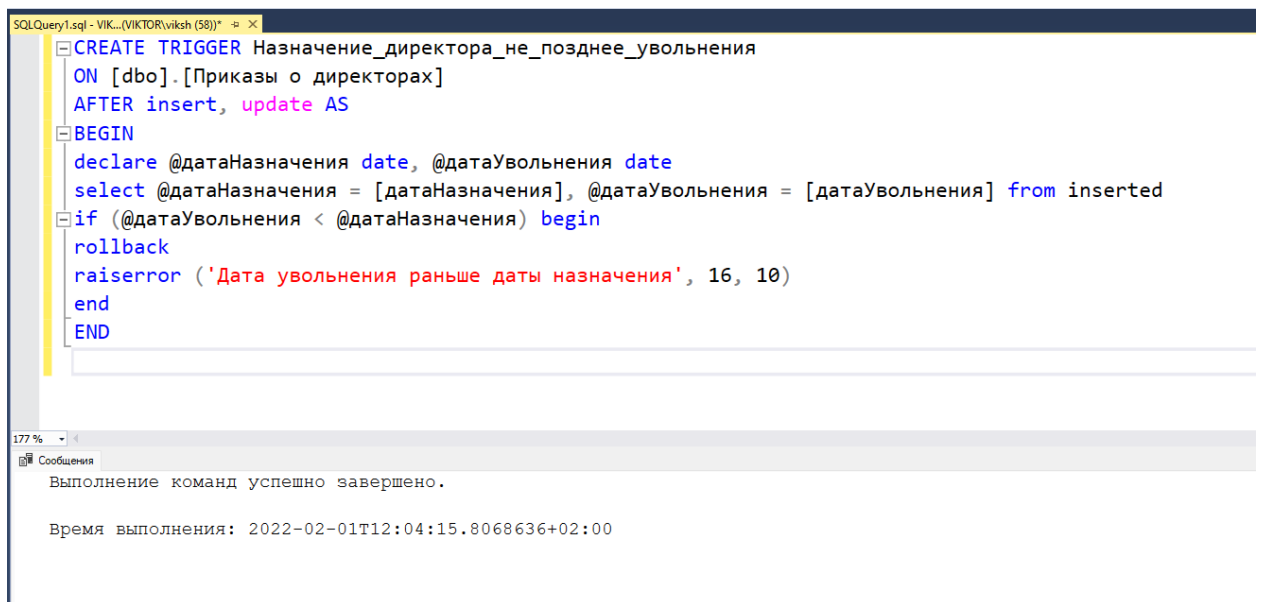
Схема БД:



Выполнение работы:

1. Дата назначение директора не позднее даты его увольнения:

```
CREATE TRIGGER Назначение_директора_не_позднее__увольнения
ON [dbo].[Приказы о директорах]
AFTER insert, update AS
BEGIN
declare @датаНазначения date, @датаУвольнения date
select @датаНазначения = [датаНазначения], @датаУвольнения = [датаУвольнения] from
inserted
if (@датаУвольнения < @датаНазначения) begin
rollback
raiserror ('Дата увольнения раньше даты назначения', 16, 10)
end
END
```



```
SQLQuery1.sql - VIKTOR\viksh (58) *
CREATE TRIGGER Назначение_директора_не_позднее__увольнения
ON [dbo].[Приказы о директорах]
AFTER insert, update AS
BEGIN
declare @датаНазначения date, @датаУвольнения date
select @датаНазначения = [датаНазначения], @датаУвольнения = [датаУвольнения] from inserted
if (@датаУвольнения < @датаНазначения) begin
rollback
raiserror ('Дата увольнения раньше даты назначения', 16, 10)
end
END
```

177 %

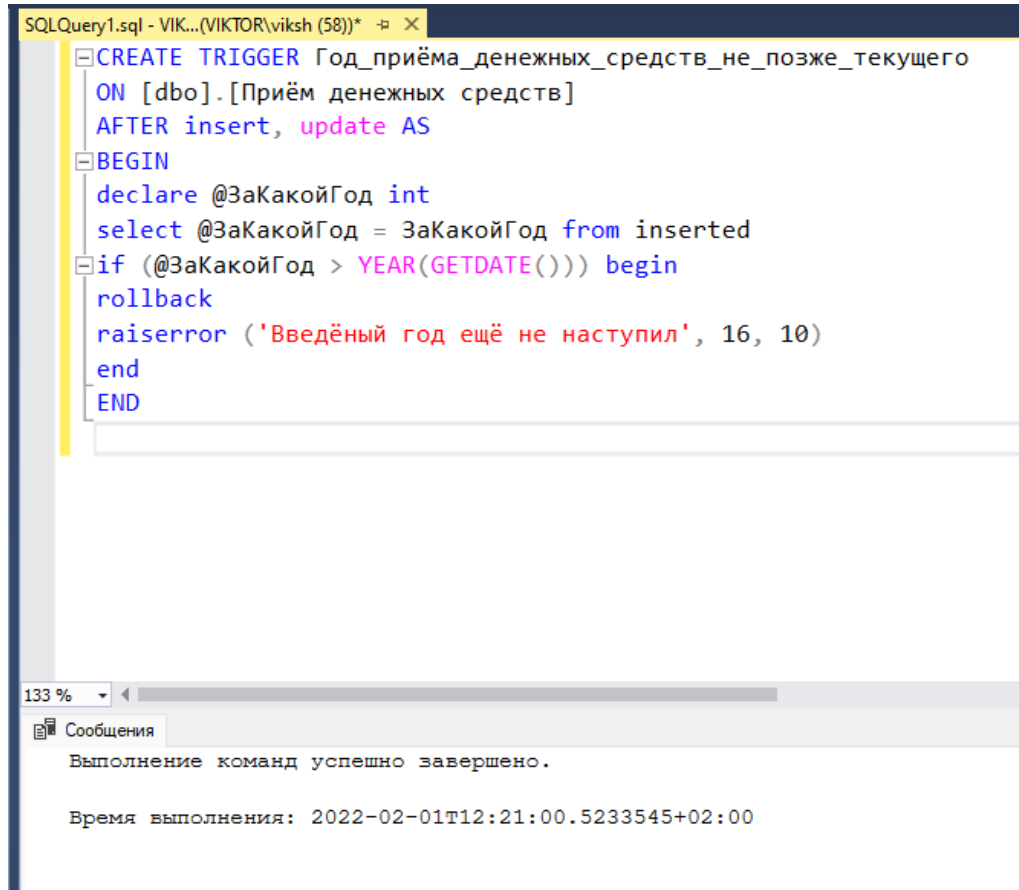
Сообщения

Выполнение команд успешно завершено.

Время выполнения: 2022-02-01T12:04:15.8068636+02:00

2. Год приёма денежных средств, не позже текущего:

```
CREATE TRIGGER Год_приёма_денежных_средств_не_позже_текущего
ON [dbo].[Приём денежных средств]
AFTER insert, update AS
BEGIN
declare @ЗаКакойГод int
select @ЗаКакойГод = ЗаКакойГод from inserted
if (@ЗаКакойГод > YEAR(GETDATE())) begin
rollback
raiserror ('Введённый год ещё не наступил', 16, 10)
end
END
```

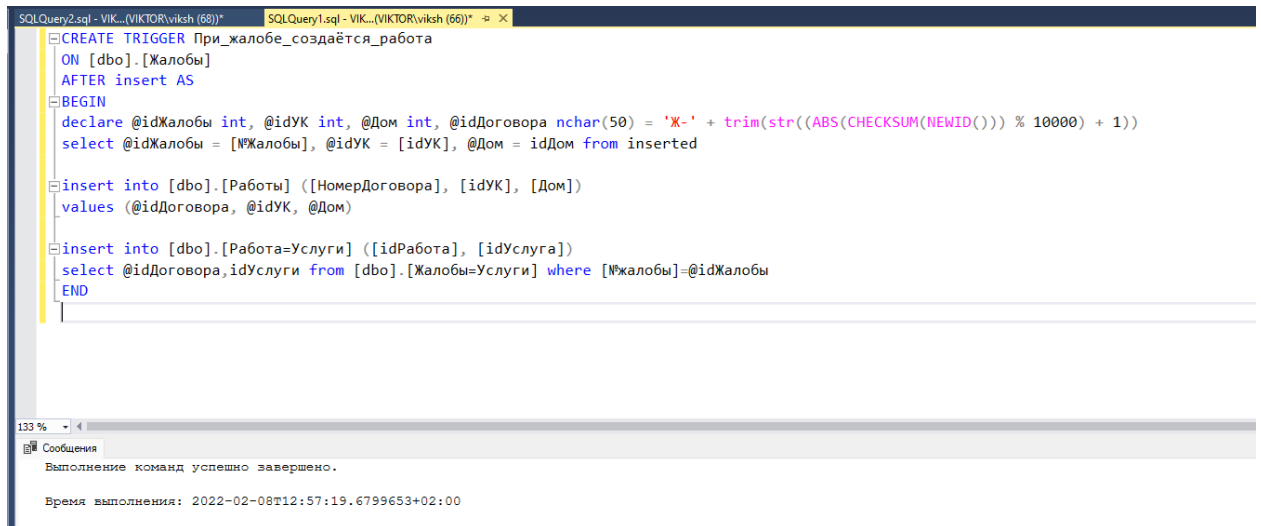


3. Если есть жалоба на услугу, создать работу

```
CREATE TRIGGER При_жалобе_создаётся_работа
ON [dbo].[Жалобы]
AFTER insert AS
BEGIN
declare @idЖалобы int, @idУК int, @Дом int, @idДоговора nchar(50) = 'Ж-' +
trim(str((ABS(CHECKSUM(NEWID())) % 10000) + 1))
select @idЖалобы = [№Жалобы], @idУК = [idУК], @Дом = idДом from inserted

insert into [dbo].[Работы] ([НомерДоговора], [idУК], [Дом])
values (@idДоговора, @idУК, @Дом)

insert into [dbo].[Работа=Услуги] ([idРабота], [idУслуга])
select @idДоговора, idУслуги from [dbo].[Жалобы=Услуги] where [№жалобы]=@idЖалобы
END
```



The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows the SQL script for creating a trigger named 'При_жалобе_создаётся_работа' on the 'Жалобы' table. The script includes variable declarations, a select statement to retrieve values from the 'inserted' table, and two insert statements into the 'Работы' and 'Работа=Услуги' tables. The bottom pane shows a message box indicating that the command was executed successfully. The execution time is listed as 2022-02-08T12:57:19.6799653+02:00.

```
SQLQuery2.sql - VIK... (VIKTOR\viksh (68))* SQLQuery1.sql - VIK... (VIKTOR\viksh (66))*
CREATE TRIGGER При_жалобе_создаётся_работа
ON [dbo].[Жалобы]
AFTER insert AS
BEGIN
declare @idЖалобы int, @idУК int, @Дом int, @idДоговора nchar(50) = 'Ж-' + trim(str((ABS(CHECKSUM(NEWID())) % 10000) + 1))
select @idЖалобы = [№Жалобы], @idУК = [idУК], @Дом = idДом from inserted

insert into [dbo].[Работы] ([НомерДоговора], [idУК], [Дом])
values (@idДоговора, @idУК, @Дом)

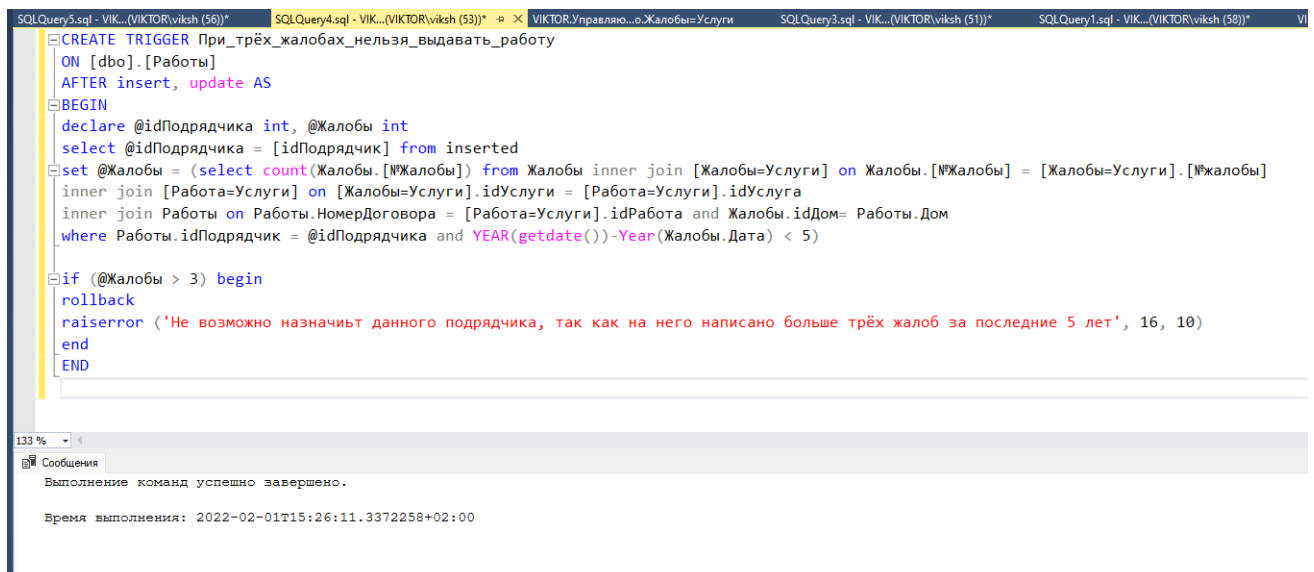
insert into [dbo].[Работа=Услуги] ([idРабота], [idУслуга])
select @idДоговора, idУслуги from [dbo].[Жалобы=Услуги] where [№жалобы]=@idЖалобы
END

Сообщения
Выполнение команд успешно завершено.
Время выполнения: 2022-02-08T12:57:19.6799653+02:00
```

4. Если есть три жалобы на подрядчика, отстранить его

```
CREATE TRIGGER При_трёх_жалобах_нельзя_выдавать_работу
ON [dbo].[Работы]
AFTER insert, update AS
BEGIN
declare @idПодрядчика int, @Жалобы int
select @idПодрядчика = [idПодрядчик] from inserted
set @Жалобы = (select count(Жалобы.[№Жалобы]) from Жалобы inner join [Жалобы=Услуги] on
Жалобы.[№Жалобы] = [Жалобы=Услуги].[№Жалобы]
inner join [Работа=Услуги] on [Жалобы=Услуги].idУслуги = [Работа=Услуги].idУслуга
inner join Работы on Работы.НомерДоговора = [Работа=Услуги].idРабота and Жалобы.idДом=
Работы.Дом
where Работы.idПодрядчик = @idПодрядчика and YEAR(getdate())-Year(Жалобы.Дата) < 5)

if (@Жалобы > 3) begin
rollback
raiserror ('Не возможно назначить данного подрядчика, так как на него написано больше
трёх жалоб за последние 5 лет', 16, 10)
end
END
```



5. Житель и жалоба

```
CREATE TRIGGER Житель_подаёт_жалобу_на_свой_дом
ON [dbo].[Жалобы]
AFTER insert, update AS
BEGIN
declare @Дом int, @idЖитель int
select @Дом = idДом, @idЖитель = idЖитель from inserted

if (@Дом != (select idДом from [Дом=Жители] where idЖитель = @idЖитель)) begin
rollback
raiserror ('Житель подал жалобу не на свой дом', 16, 10)
end

END
```

