**Лабораторная работа № 7. Использование представлений**

**КУРНОСЕНКО СОФЬЯ**

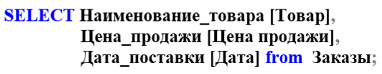
**Представление** (View) – это объект базы данных, представляющий собой *поименованный* SELECT-запрос, который хранится в базе данных. Представление создается с помощью оператора CREATE, удаляется с помощью оператора DROP и изменяется с помощью ALTER.

1.Разработать представление с именем **Преподаватель**. Представление должно быть построено на основе SELECT-запроса к таблице **TEACHER** и содержать следующие столбцы: **код** (**TEACHER**), **имя преподавателя** (**TEACHER\_NAME**), **пол** (**GENDER**), **код кафедры** (**PULPIT**).

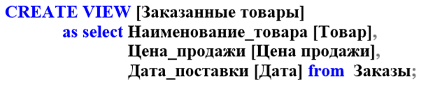
***→ Теория (методичка)***

При создании представления к SELECT-запросу предъявляются следующие требования: секцию ORDER BY можно использовать только совместно с опцией TOP; не допускается применение секции INTO, COMPUTE и COMPUTE BY; все столбцы результирующего набора должны быть поименованы.

Пусть требуется определить наименования заказанных товаров, цены их продаж и даты поставки:



Можно создать представление с именем **Заказанные товары**:



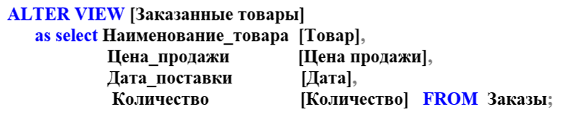
Тогда тот же самый запрос записывается короче:

**SELECT \* from [Заказанные товары]**

Это представление можно использовать и в других запросах, например:

**SELECT \* from [Заказанные товары] order by [Дата] ;**

Чтобы изменить представление, надо использовать оператор ALTER:



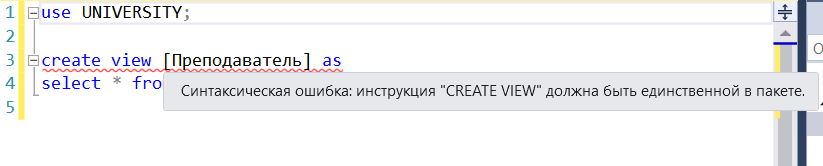
Удаляется представление с помощью оператора DROP:

**DROP VIEW [Заказанные товары];**

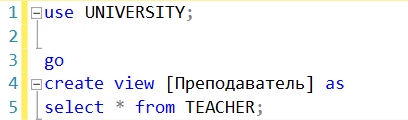
***→ Выполнение заданий***

**Инструкция CREATE VIEW должна быть единственной инструкцией пакета.** (Это означает, что эту инструкцию следует отделять от других инструкций группы посредством инструкции GO.)

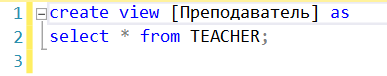
Ошибка, потому что инструкция create view не отделена от других:



Теперь все нормально:



Или можно так:

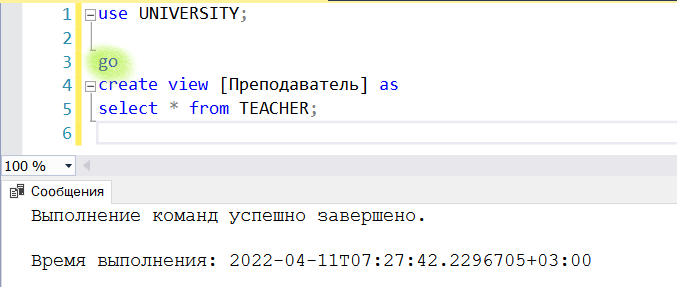


use UNIVERSITY;

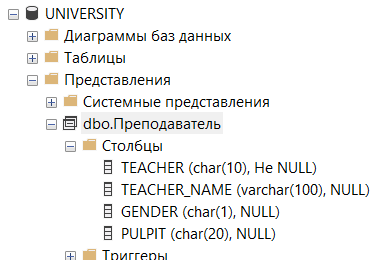
go

create view [Преподаватель] as

select \* from TEACHER;



В БД:

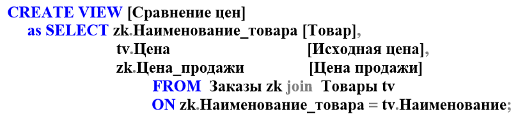


2.Разработать и создать представление с именем **Количество кафедр**. Представление должно быть построено на основе SELECT-запроса к таблицам **FACULTY** и **PULPIT**.

Представление должно содержать следующие столбцы: **факультет** (**FACULTY.FACULTY\_ NAME**), **количество кафедр** (вычисляется на основе строк таблицы **PULPIT**).

***→ Теория (методичка)***

Представление **Сравнение цен** выводит информацию об исходных ценах и ценах продажи тех товаров, которые заказаны клиентами, т. е. содержатся в таблице **Заказы**:



При создании представлений, позволяющих выполнять операции INSERT, DELETE и UPDATE, базовый SELECT-запрос должен удовлетворять правилам:

− запрос не должен содержать секцию группировки GROUP BY;

− запрос не должен применять агрегатные функции, опции DISTINCT и TOP, операторы UNION, INTERSECT и EXCEPT;

− в SELECT-списке запроса не должно быть вычисляемых значений;

− в секции FROM запроса должна указываться только одна таблица.

В приведенном выше примере представление не удовлетворяет одному из этих правил.

***→ Выполнение заданий***

go

create view [Количество кафедр] as

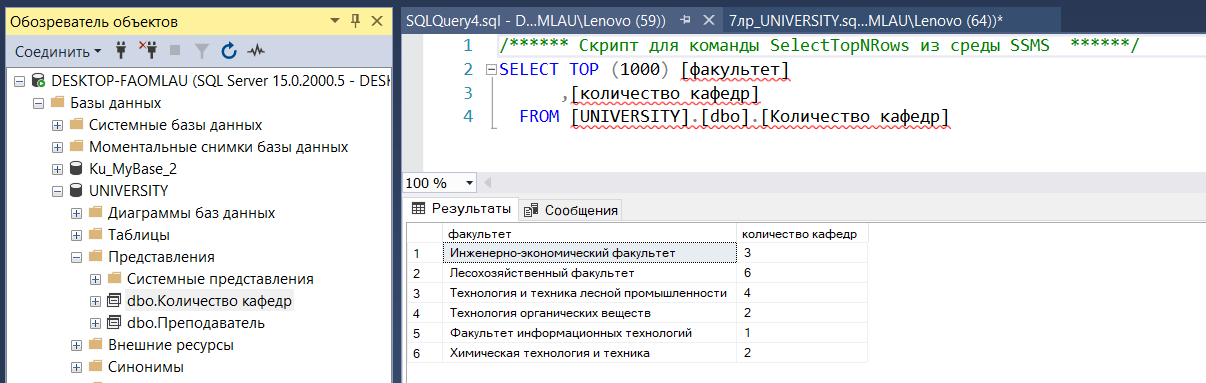
select FACULTY.FACULTY\_NAME as [факультет],

count(PULPIT.PULPIT) as [количество кафедр]

from FACULTY inner join PULPIT

on FACULTY.FACULTY = PULPIT.FACULTY

group by FACULTY.FACULTY\_NAME;

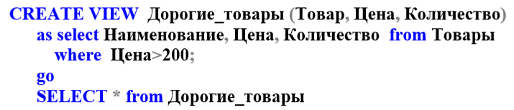


3.Разработать и создать представление с именем **Аудитории**. Представление должно быть построено на основе таблицы **AUDITORIUM** и содержать столбцы: **код** (**AUDITORIUM**), **наименование аудитории** (**AUDITORIUM\_NAME**)**.**

Представление должно отображать только лекционные аудитории (в столбце **AUDITORIUM\_ TYPE** строка, начинающаяся с символа **ЛК**) и допускать выполнение оператора INSERT, UPDATE и DELETE.

***→ Теория (методичка)***

Имя представления может содержать параметры, которые заключаются в скобки и отображаются в первой строке результирующего набора.



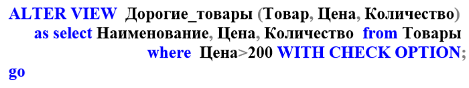
Операторы INSERT осуществят вставку строк с новой информацией:

**INSERT Дорогие\_товары values('Диван', 300, 3)**

**INSERT Дорогие\_товары values('Шкаф', 150, 7)**

Чтобы операция вставки не могла осуществиться в том случае, когда информация не удовлетворяет условию, записанному в секции Where, то следует создавать представление с опцией WITH CHECK OPTION.

Например, можно изменить представление **Дорогие\_товары**:



Тогда оператор INSERT не выполнится, поскольку цена не удовлетворяет нужному условию секции Where.

**INSERT Дорогие\_товары values('Стол', 80, 9)**

***→ Выполнение заданий***

go

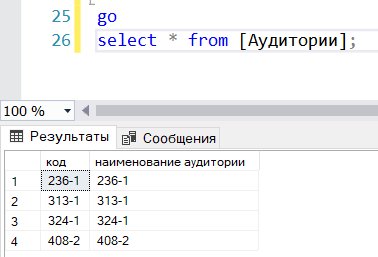
create view [Аудитории]([код], [наименование аудитории ]) as

select AUDITORIUM,

AUDITORIUM\_NAME

from AUDITORIUM

where AUDITORIUM\_TYPE like 'ЛК%';



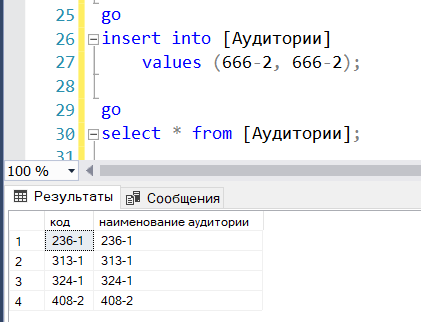
Мы наложили условие *where AUDITORIUM\_TYPE like 'ЛК%'* на выборку для представления (без использования *with check option*). Попробуем добавить новую строку в представление:

go

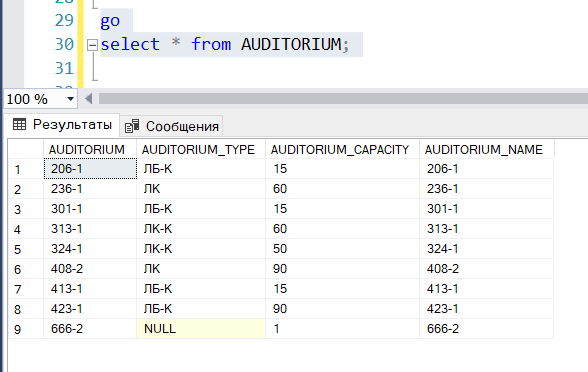
insert into [Аудитории]

values ('666-2', '666-2');

Строка добавлена, теперь выведем представление и увидим, что там добавленная строка не отображается:



Однако в саму таблицу строка попала, в качестве неуказанных значений столбцов, получив значения столбцов по умолчанию:



4.Разработать и создать представление с именем **Лекционные\_аудитории**.

Представление должно быть построено на основе SELECT-запроса к таблице **AUDITORIUM** и содержать следующие столбцы: **код** (**AUDITORIUM**), **наименование аудитории** (**AUDITORIUM\_NAME**)**.**

Представление должно отображать только лекционные аудитории (в столбце **AUDITORIUM\_TYPE** строка, начинающаяся с символов **ЛК**).

Выполнение INSERT и UPDATE допускается, но с учетом ограничения, задаваемого опцией WITH CHECK OPTION.

***→ Выполнение заданий***

go

create view Лекционные\_аудитории as

select AUDITORIUM,

AUDITORIUM\_NAME

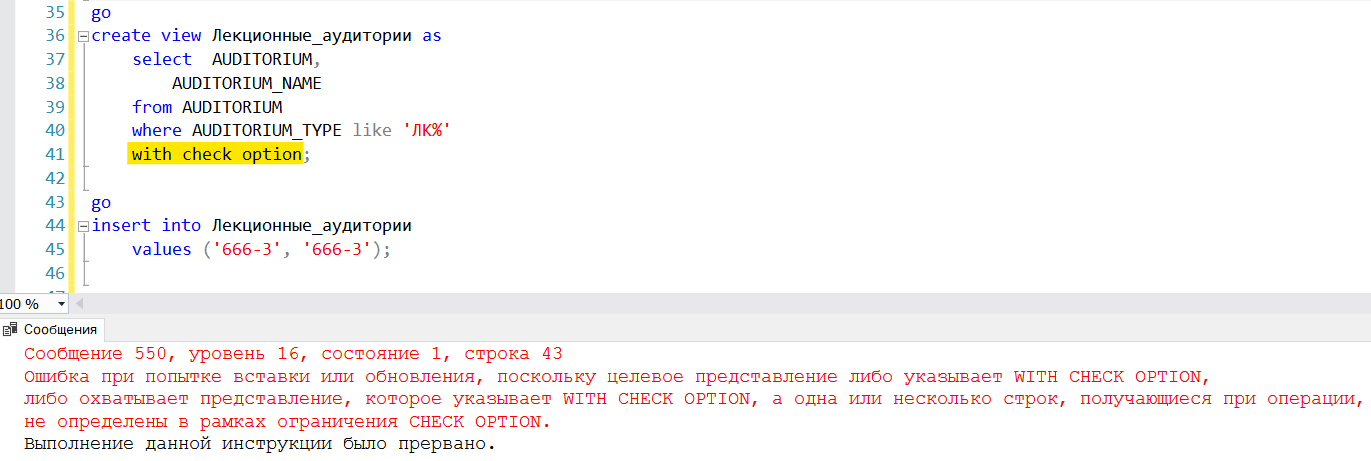
from AUDITORIUM

where AUDITORIUM\_TYPE like 'ЛК%'

with check option;

При модифицировании строк представления также модифицируется содержимое таблицы в его основе.

Опция *with check option* делает невозможной изменение представления (а соответственно и таблицы, на основе которой представление было создано), если значения не соответствуют условию оператора *where*. Так как у нас представление не содержит столбца AUDITORIUM\_TYPE, на который налагается условие, то мы вообще не можем внести изменения в представление (т. е. не можем использовать *update* и *insert*):

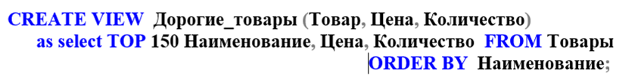


Т.о. *with check option* обеспечивает то, что если строка не может попасть в само представление из-за несоответствия условию добавления в это представление, то и в саму таблицу эта строка не будет добавлена.

5.Разработать представление с именем **Дисциплины**. Представление должно быть построено на основе SELECT-запроса к таблице **SUBJECT**, отображать все дисциплины в алфавитном порядкеи содержать следующие столбцы: **код** (**SUBJECT**), **наименование дисциплины** (**SUBJECT\_NAME**) и **код кафедры** (**PULPIT**). Использовать TOP и ORDER BY.

***→ Теория (методичка)***

Поскольку секцию ORDER BY можно использовать только совместно с опцией TOP, то представление может выглядеть следующим образом:



***→ Выполнение заданий***

go

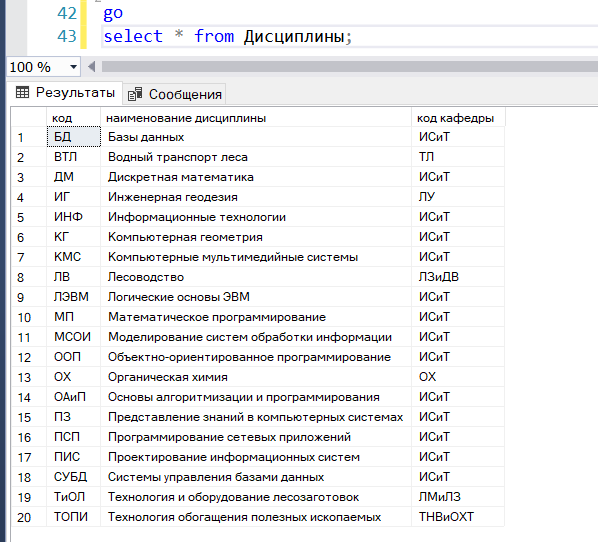
create view Дисциплины(код, [наименование дисциплины], [код кафедры]) as

select top 20

SUBJECT, SUBJECT\_NAME, PULPIT

from SUBJECT

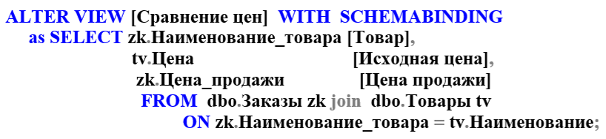
order by SUBJECT\_NAME;

******

6.Изменить представление **Количество\_кафедр**, созданное в задании 2 так, чтобы оно было привязано к базовым таблицам. Продемонстрировать свойство привязанности представления к базовым таблицам. Примечание: использовать опцию SCHEMABINDING.

***→ Теория (методичка)***

Опция SCHEMABINDING устанавливает запрещение на операции с таблицами и представлениями, которые могут привести к нарушению работоспособности представления.



При использовании опции SCHEMABINDING требуется использовать в SELECT-запросе для имен таблиц и представлений двухкомпонентный формат (в имени присутствует наименование схемы).

***→ Выполнение заданий***

-- 6

go

alter view dbo.[Количество кафедр] with schemabinding

as

select FACULTY.FACULTY\_NAME as [факультет],

count(PULPIT.PULPIT) as [количество кафедр]

from dbo.FACULTY inner join dbo.PULPIT

on FACULTY.FACULTY = PULPIT.FACULTY

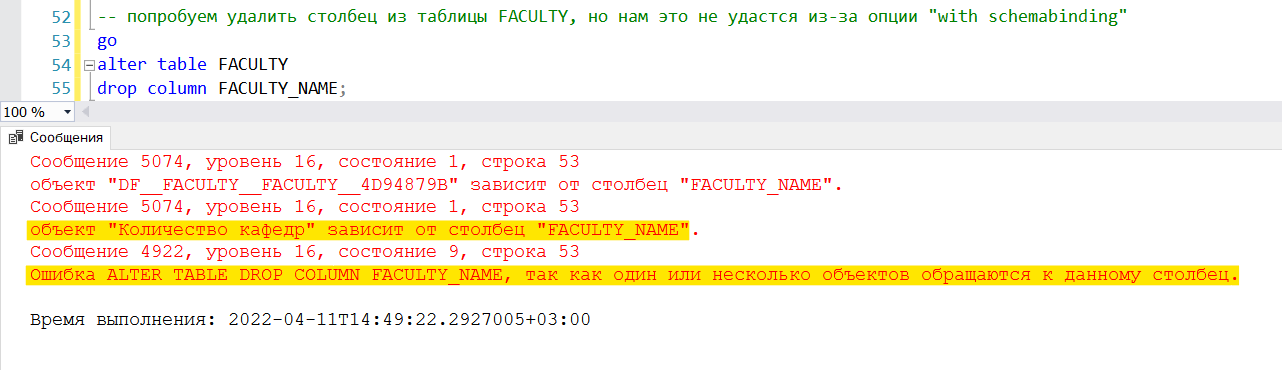
group by FACULTY.FACULTY\_NAME;

-- попробуем удалить столбец из таблицы FACULTY, но нам это не удастся из-за опции "with schemabinding"

go

alter table FACULTY

drop column FACULTY\_NAME;

******

7. Разработать представления для базы данных **X\_MyBASE**.

use Ku\_MyBase\_2;

-- 1

go

create view [Представление маршрутов] as

select \* from Маршруты;

-- 2 Представление соединения таблиц "Водители" и "Водители на перевозках", отобразить Код\_водителя, ФИО водителя и Код\_перевозки

go

create view [Водители\_перевозки] as

select Водители.Код\_водителя,

Водители.Фамилия,

Водители.Имя,

Водители.Отчество,

[Водители на перевозках].Код\_перевозки

from Водители inner join [Водители на перевозках]

on Водители.Код\_водителя = [Водители на перевозках].Код\_водителя;

-- 3 Представление с водителями со стажем от 5 лет, должно допускать выполнение оператора INSERT, UPDATE и DELETE

go

create view [Водители со стажем от 5 лет] as

select \* from Водители

where Стаж >= 5;

-- представление пункта 3 допускает использование INSERT, UPDATE и DELETE

-- данная строка не соответствует условию "Стаж >= 5", поэтому не попадет в представление, зато появится в таблице

go

insert [Водители со стажем от 5 лет] values('Томина', 'Анастасия', 'Владимировна', 4);

-- 4 переписать представление пункта 3 с опцией WITH CHECK OPTION

go

create view [Водители со стажем от 5 лет с проверкой] as

select \* from Водители

where Стаж >= 5

WITH CHECK OPTION;

-- данная строка не соответствует условию "Стаж >= 5", поэтому не попадет в представление

-- а так как в представлении указана опция WITH CHECK OPTION, то эта строка и в таблицу не попадет

-- выкинет ошибку

go

insert [Водители со стажем от 5 лет с проверкой] values('Томин', 'Алексей', 'Никитич', 4);

-- 5

go

create view [Представление маршрутов с сортировкой] as

select TOP 10 \* from Маршруты

order by Название;

-- 6

go

alter view dbo.[Водители\_перевозки]

with schemabinding

as

select Водители.Код\_водителя,

Водители.Фамилия,

Водители.Имя,

Водители.Отчество,

[Водители на перевозках].Код\_перевозки

from dbo.Водители inner join dbo.[Водители на перевозках]

on Водители.Код\_водителя = [Водители на перевозках].Код\_водителя;