

Частное учреждение образования
«Колледж бизнеса и права»

УТВЕРЖДАЮ
Заведующий
методическим кабинетом
_____ Е.В. Фалей
«__» _____ 2017

Специальность: «Программное обеспечение информационных технологий»	Дисциплина: «Базы данных и системы управления базами данных»
Составлена на основании учебной программы, утвержденной директором Колледжа бизнеса и права 30.12.2016	

Лабораторная работа № 18
Инструкционно-технологическая карта

Тема: Создание и модификация представлений в СУБД

Цель работы:

- получить навыки работы с представлениями посредством основных инструкций: CREATE VIEW, ALTER VIEW, DROP VIEW;
- изучить инструкции DML (SELECT, INSERT, UPDATE, DELETE) применительно к представлениям, их ограничения при применении.

Время выполнения: 2 часа

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить разделы лекционного курса по теме «Описание данных на основе SQL» и теме «Триггеры и хранимые процедуры».
2. Получить у преподавателя индивидуальное задание и выполнить лабораторную работу в соответствии с вариантом задания согласно описанной в разделе «Пример выполнения работы» методике настоящей инструкционно-технологической карты.
3. Ответить на контрольные вопросы.

1. ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ

1. Создание и модификация представлений в СУБД SQL Server

Представления – это объекты базы данных, которые всегда создаются на основе одной или более базовых таблиц (или других представлений), используя информацию метаданных. Эта информация (включая имя представления и способ получения строк из базовых таблиц) — все, что сохраняется физически для представления. По этой причине представления также называются виртуальными таблицами.

Данные базовой таблицы хранятся на диске. В отличие от базовых таблиц, представления по умолчанию не существуют физически, т. е. их содержимое не сохраняется на диске.

Представление создается посредством инструкции CREATE VIEW, синтаксис которой выглядит следующим образом:

```
CREATE VIEW view_name[(column_list)]
[WITH {ENCRYPTION SCHEMABINDING | VIEW_METADATA}]
AS select_statement
[WITH CHECK OPTION]
```

Замечание 1. Инструкция CREATE VIEW должна быть единственной инструкцией пакета, т.е. эту инструкцию следует отделять от других инструкций группы посредством инструкции GO.

Параметр	Значение параметра
view_name	имя представления
column_list	необходимо указать список имен, которые будут использоваться в качестве имен столбцов представления. Если необязательный параметр опущен, то используются имена столбцов таблиц, по которым создается представление
select_statement	необходимо задать инструкцию SELECT, которая извлекает строки и столбцы из таблиц (или других представлений). См. замечание 2
WITH ENCRYPTION	задает шифрование инструкции SELECT, повышая таким образом уровень безопасности системы баз данных. См. замечание 3
SCHEMABINDING	привязывает представление к схеме таблицы, по которой оно создается
VIEW_METADATA	предоставляет возможность обновления всех столбцов представления (за исключением столбцов с типом данных TIMESTAMP), если представление имеет триггеры INSERT или UPDATE INSTEAD OF

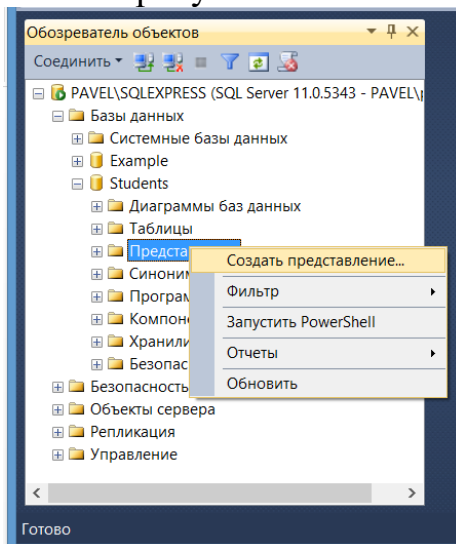
Замечание 2. Инструкция SELECT в представлении не может содержать предложение ORDER BY или параметр INTO. Кроме этого, по временным таблицам нельзя выполнять запросы.

Замечание 3. Предложение SCHEMABINDING привязывает представление к схеме таблицы, по которой оно создается. Когда это предложение указывается, имена объектов баз данных в инструкции SELECT должны состоять из двух частей, т. е. в виде Owner.db_object, где owner — владелец, а db_object может быть таблицей, представлением или определяемой пользователем функцией. Любая попытка модифицировать структуру представлений или таблиц, на которые ссылается созданное таким образом представление, будет неудачной. Чтобы такие таблицы или представления можно было модифицировать (инструкцией ALTER) или удалять (инструкцией DROP), нужно удалить это представление или убрать из него предложение SCHEMABINDING.

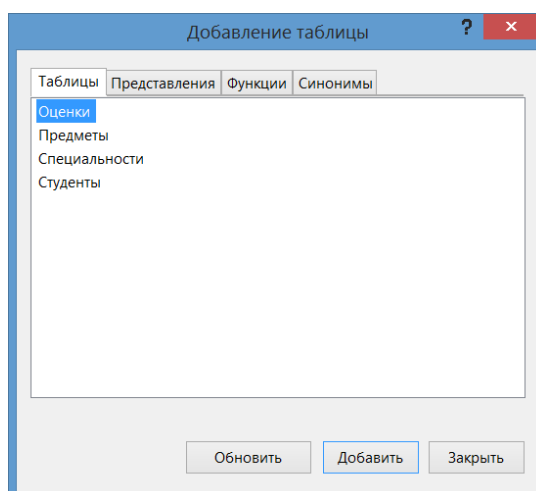
Представления можно использовать для следующих целей:

- ✓ для ограничения использования определенных столбцов и/или строк таблиц, т.е. представления можно использовать для управления доступом к определенной части одной или нескольких таблиц;
- ✓ для скрытия подробностей сложных запросов. Если для приложения базы данных требуются запросы со сложными операциями соединения, создание соответствующих представлений может упростить такие запросы;
- ✓ для ограничения вставляемых или обновляемых значений некоторым диапазоном.

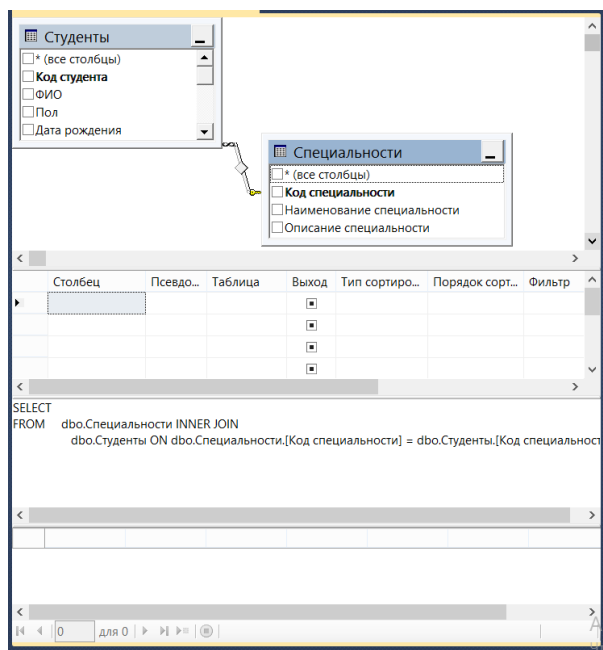
Представления можно создавать посредством среды Management Studio. Для этого необходимо выбрать нужную базу данных в обозревателе объектов, щелкнуть в ней правой кнопкой мыши узел View (Представления) и в открывшемся контекстном меню необходимо выбрать New View (Новой представление), как показано на рисунке ниже.



В результате выполнения указанных выше действий откроется окно выбора базовых таблиц, необходимых для создания представления, как показано ниже:

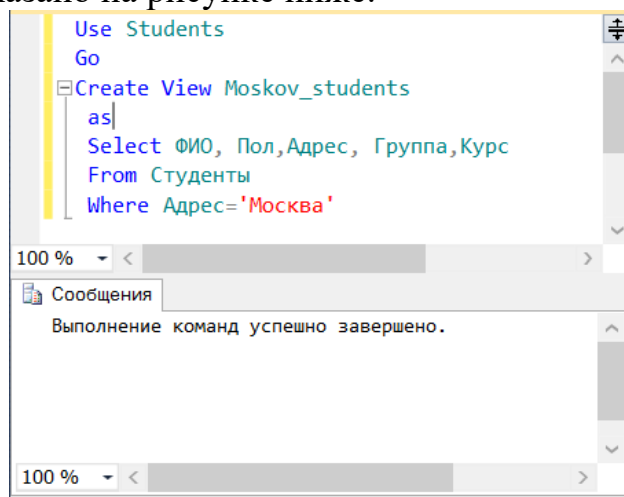


После выбора базовых таблиц откроется редактор представлений, который представлен на рисунке ниже:



В редакторе представлений можно выбрать еще раз базовые таблицы и строки в этих таблицах для создания представлений. Кроме того, можно присвоить представлению имя и определить условия в предложении WHERE соответствующего запроса.

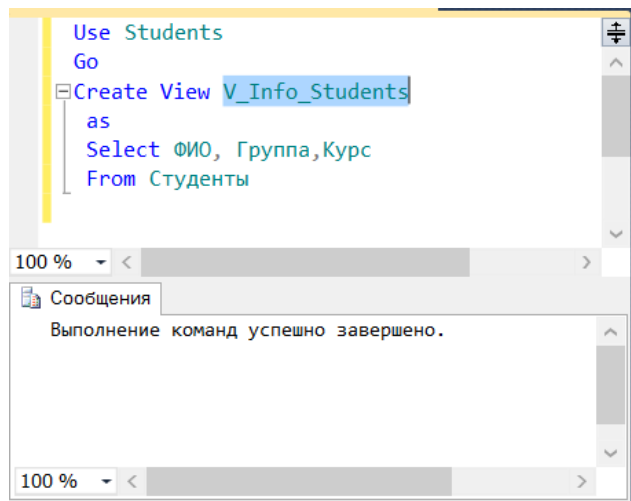
В следующем примере показано создание представления Moskov_students, которое определяется строками и столбцами, возвращаемыми запросом на выборку студентов, проживающих в Москве. Представление показано на рисунке ниже.



Результат можно просмотреть в узле **View (Представления)**, далее Moskov_students → «Изменить первые 200 строк» или с помощью инструкции SELECT (см. ниже выборку из представления):

	ФИО	Пол	Адрес	Группа	Курс
▶	Иванов А.И.	Мужской	Москва	ММ11	1
	Петрова И....	Женский	Москва	ПИ21	2
*	NULL	NULL	NULL	NULL	NULL

Запрос в следующем примере выбирает для включения в представление V_Info_Students информацию о группе и курсе студента из таблицы Студенты.



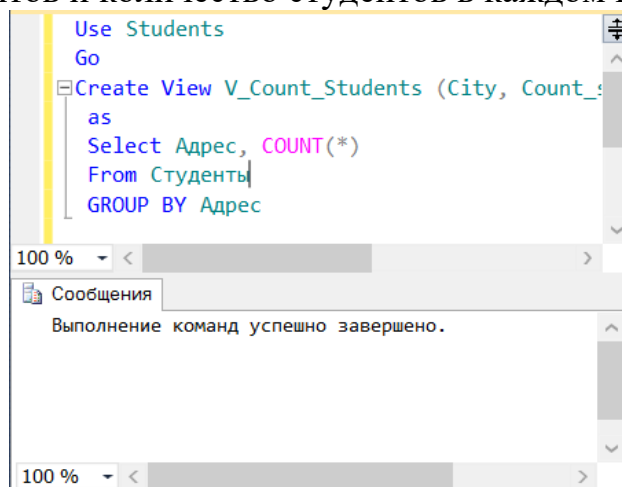
Результат представлен на рисунке ниже:

ФИО	Группа	Курс
Иванов А.И.	ММ11	1
Петрова И...	ПИ21	2
Мухин М.А.	СТ22	3
Сидорова ...	МО31	4
Кожевнико...	БУ33	5
Пальчиков...	ММ12	1
Царегород...	ПИ41	2
Баранова Г...	СТ42	3
Леухин П.Г.	МО51	4
Николаева ...	БУ53	5
NULL	NULL	NULL

В общем формате инструкции CREATE VIEW не обязательно указывать имена столбцов представления. Однако, с другой стороны, в приведенных далее двух случаях обязательно требуется явно указывать имена столбцов:

1. если столбец представления создается из выражения или агрегатной функции;
2. если два или больше столбцов представления имеют одинаковое имя в базовой таблице.

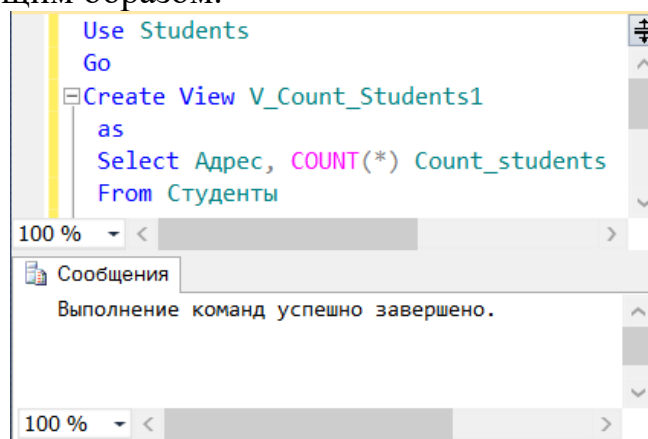
В следующем примере показано создание представления, для которого явно указываются имена столбцов. В представление включены города проживания студентов и количество студентов в каждом городе.



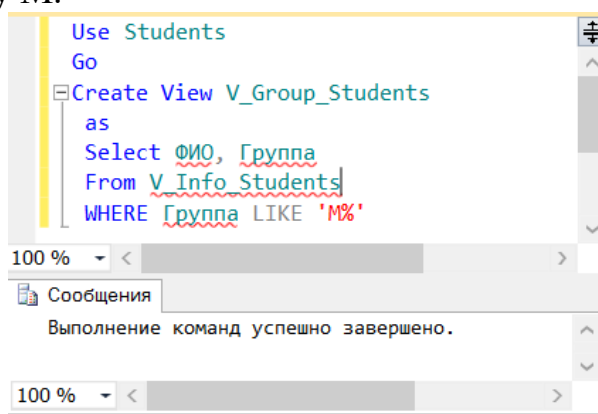
Результат представлен на рисунке ниже:

	City	Count_stud...
▶	Казань	2
	Москва	2
	Самара	2
	Саратов	2
	Чебоксары	1
	Челябинск	1
*	NULL	NULL

Не требуется явно указывать список столбцов в инструкции CREATE VIEW, если применить заглавия столбцов. Тогда предыдущий пример будет выглядеть следующим образом.



Представление можно создавать из другого представления. Следующее представление V_Group_Students создается из представления V_Info_Students: выбираются все студенты, у которых название группы начинается на букву М.



Результат представлен на рисунке ниже:

	ФИО	Группа
▶	Сидорова В.К	МО31
	Пальчиков...	ММ12
	Леухин П.Г.	МО51
*	NULL	NULL

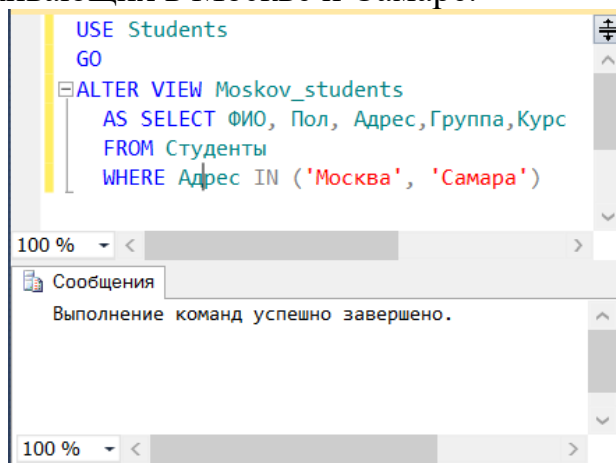
2. Изменение и удаление представлений

Для изменения определения представления в языке Transact-SQL применяется инструкция ALTER VIEW. Синтаксис этой инструкции

аналогичен синтаксису инструкции CREATE VIEW, применяющейся для создания представления.

Использование инструкции ALTER VIEW позволяет избежать переназначения существующих разрешений для представления. Кроме этого, изменение представления посредством этой инструкции не влияет на объекты базы данных, зависящие от этого представления. Если же модифицировать представление, сначала удалив его (инструкция DROP VIEW), а затем создав новое представление с требуемыми свойствами (инструкция CREATEVIEW), то все объекты базы данных, которые ссылаются на это представление, не будут работать должным образом, по крайней мере, в период времени между удалением представления и его воссоздания.

В следующем примере инструкция ALTER VIEW расширяет инструкцию SELECT в представлении Moskov_students новым условием: вывод студентов, проживающих в Москве и Самаре.

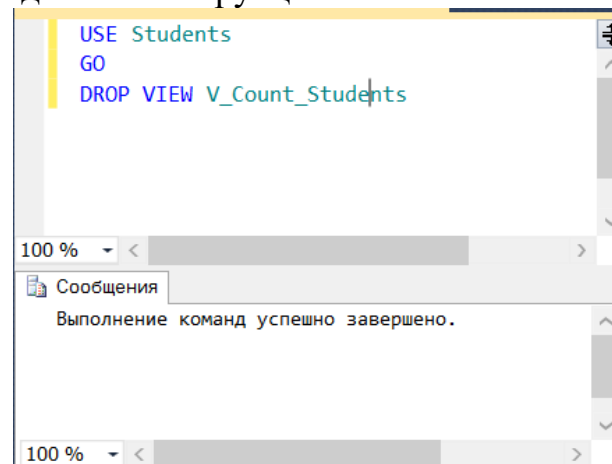


Результат представлен на рисунке ниже:

ФИО	Пол	Адрес	Группа	Курс
Иванов А.И.	Мужской	Москва	ММ11	1
Петрова И...	Женский	Москва	ПИ21	2
Мухин М.А.	Мужской	Самара	СТ22	3
Царегород...	Мужской	Самара	ПИ41	2
* NULL	NULL	NULL	NULL	NULL

Инструкция DROP VIEW удаляет из системных таблиц определение указанного в ней представления.

В следующем примере показано удаление представления V_Count_Students посредством инструкции DROP VIEW.



При удалении представления инструкцией DROP VIEW все другие представления, основанные на удаленном, также удаляются.

Замечание 4. При удалении базовой таблицы представления, основанные на ней, не удаляются автоматически. Это означает, что все представления для удаленной таблицы нужно удалять явно, используя инструкцию DROP VIEW. С другой стороны, представления удаленной таблицы можно снова использовать на новой таблице, имеющей такую же логическую структуру, как и удаленная.

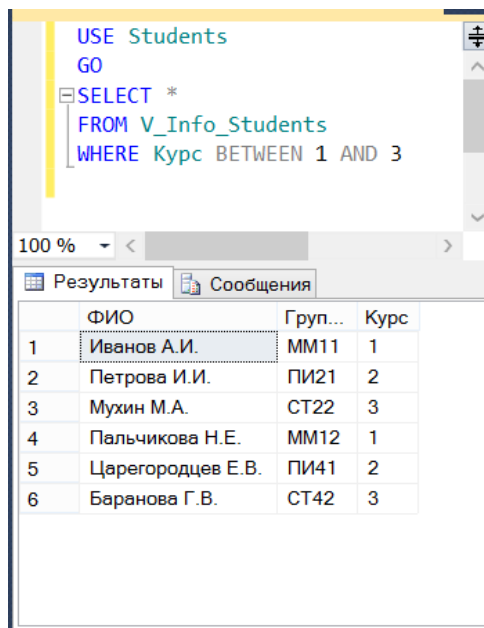
3. Инструкции DML и представления

Выборка информации из представлений и ее модификация осуществляются посредством тех же самых инструкций языка Transact-SQL, что и для выборки и модификации информации из базовых таблиц.

4. Выборка информации из представления

Для всех практических целей представление это то же самое, что и любая базовая таблица базы данных. Поэтому выборку информации из представления можно рассматривать, как преобразование инструкций запроса по представлению в эквивалентные операции по базовым таблицам, на основе которых создано это представление.

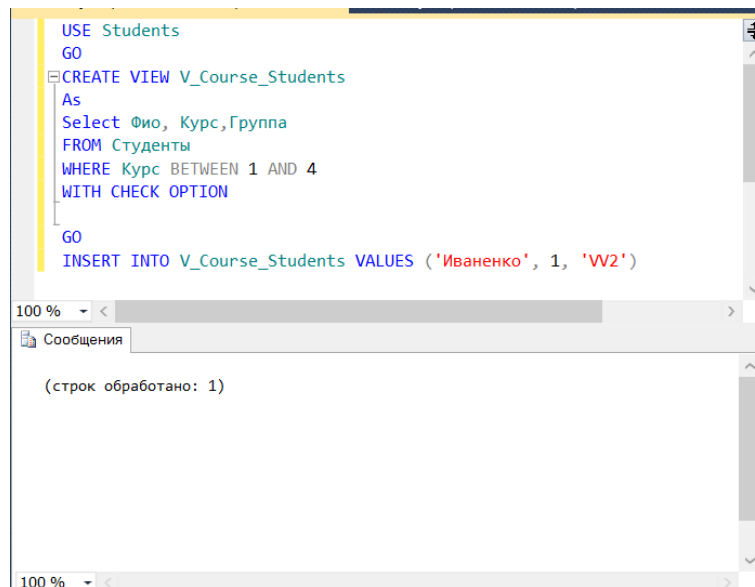
В следующем примере из представления V_Info_Students выбираются все студенты с 1 по 3 курс. Выборка и ее результат представлены на рисунке ниже.



5. Инструкция INSERT и представление

Инструкцию INSERT можно применять с представлением, как если бы оно было обычной базовой таблицей. Вставляемые в представление строки в действительности вставляются в таблицу в основе представления.

В следующем примере создается представление V_Course_Students, в которое включаются все студенты с 1 по 4 курс

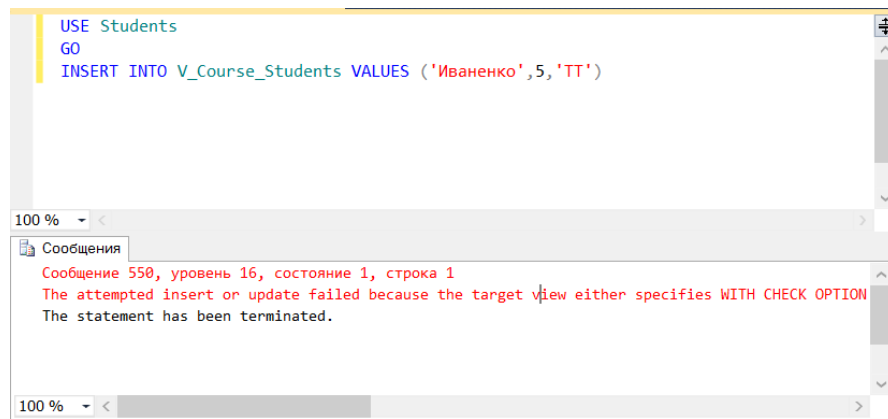


Результат представлен на рисунке ниже:

	Фιο	Курс	Группа
►	Иванов А.И.	1	ММ11
	Петрова И....	2	ПИ21
	Мухин М.А.	3	СТ22
	Сидорова ...	4	МО31
	Пальчиков...	1	ММ12
	Царегород...	2	ПИ41
	Баранова Г...	3	СТ42
	Леухин П.Г.	4	МО51
	Иваненко	1	VV2
*	NULL	NULL	NULL

При использовании представления обычно возможно вставить строку, которая не удовлетворяет условиям в предложении WHERE запроса представления. Чтобы ограничить вставку только строками, которые удовлетворяют условиям запроса, применяется предложение WITH CHECK OPTION. При использовании этого предложения SQL Server проверяет каждую вставляемую строку на удовлетворение условий предложения WHERE. Если это предложение отсутствует, такая проверка не выполняется, вследствие чего каждая вставляемая в представление строка также вставляется в таблицу в его основе. Это может вызвать путаницу, когда строка вставляется в представление, но впоследствии не возвращается из этого представления инструкцией SELECT, т. к. для нее принудительно выполняются условия предложения WHERE. Предложение WITH CHECK OPTION также применяется и с инструкцией UPDATE.

В следующем примере система проверяет, соответствует ли вставляемое в столбец КУРС значение условию в предложении WHERE инструкции SELECT представления V_Course_Students. Если вставляемое значение не удовлетворяет условиям, строка не вставляется, как показано на рисунке ниже.



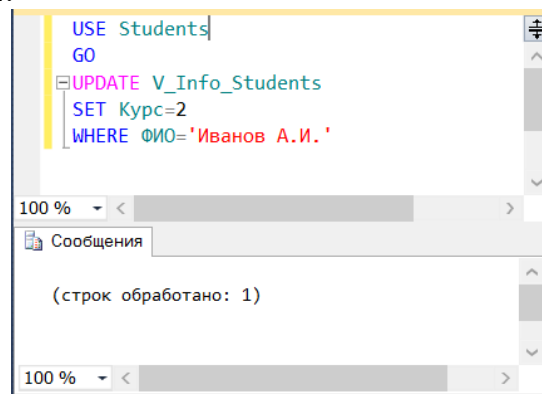
Вставку строк в таблицу, на которой основано представление, нельзя выполнить, если это представление содержит одну из следующих возможностей:

- ✓ предложение FROM в определении представления содержит более чем одну таблицу, и список столбцов содержит столбцы более чем из одной таблицы;
- ✓ столбец в представлении создается из агрегатной функции;
- ✓ инструкция SELECT в представлении содержит предложение GROUP BY или параметр DISTINCT;
- ✓ столбец в представлении создается из константы или выражения.

6. Инструкция UPDATE и представление

Инструкцию UPDATE можно применять с представлением, как будто бы это была базовая таблица. При модифицировании строк представления также модифицируется содержимое таблицы в его основе.

В следующем примере происходит обновление таблицы Студенты посредством ее представления V_Info_Students: изменяется курс студенту Иванову А.И. с 1 на 2.



Содержимое представления V_Info_Students до выполнения операции UPDATE представлено на рисунке ниже:

	ФИО	Группа	Курс
▶	Иванов А.И.	ММ11	1
	Петрова И....	ПИ21	2
	Мухин М.А.	СТ22	3
	Сидорова ...	МО31	4
	Кожевнико...	БУ33	5
	Пальчиков...	ММ12	1
	Царегород...	ПИ41	2
	Баранова Г...	СТ42	3
	Леухин П.Г.	МО51	4
	Николаева ...	БУ53	5
	Иваненко	ВВ2	1
	Иваненко	ТТ	2
*	NULL	NULL	NULL

Результат выполнения операции UPDATE представлен на рисунке ниже:

	ФИО	Группа	Курс
▶	Иванов А.И.	ММ11	2
	Петрова И....	ПИ21	2
	Мухин М.А.	СТ22	3
	Сидорова ...	МО31	4
	Кожевнико...	БУ33	5
	Пальчиков...	ММ12	1
	Царегород...	ПИ41	2
	Баранова Г...	СТ42	3
	Леухин П.Г.	МО51	4
	Николаева ...	БУ53	5
	Иваненко	ВВ2	1
	Иваненко	ТТ	2
*	NULL	NULL	NULL

Логическое значение предложения WITH CHECK OPTION для инструкции UPDATE имеет такое же значение, как и для инструкции INSERT.

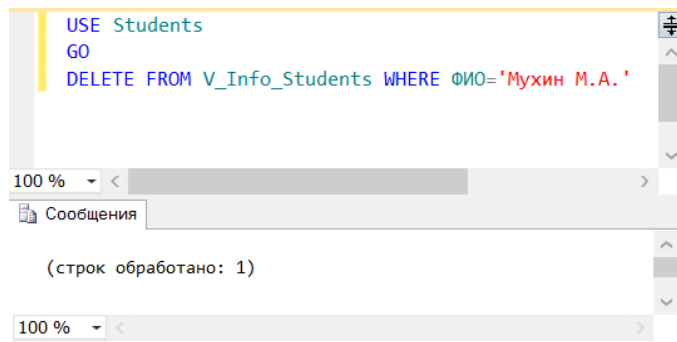
Модификацию столбцов таблицы, на которой основано представление, нельзя выполнить, если это представление содержит одну из следующих возможностей:

- ✓ предложение FROM в определении представления включает более чем одну таблицу, и список столбцов содержит столбцы из более чем одной таблицы;
- ✓ столбец представления создается из агрегатной функции;
- ✓ инструкция SELECT в представлении содержит предложение GROUP BY или параметр DISTINCT;
- ✓ столбец в представлении создается из константы или выражения.

7. Инструкция DELETE и представление

С помощью представления можно удалить строки из таблицы, на которой оно основано.

В следующем примере происходит удаление строки таблицы Студенты посредством ее представления V_Info_Students: удаляется информация о студенте Мухин М.А.



Содержимое представления V_Info_Students до выполнения операции DELETE представлено на рисунке ниже:

	ФИО	Группа	Курс
▶	Иванов А.И.	ММ11	1
	Петрова И....	ПИ21	2
	Мухин М.А.	СТ22	3
	Сидорова ...	МО31	4
	Кожевнико...	БУ33	5
	Пальчиков...	ММ12	1
	Царегород...	ПИ41	2
	Баранова Г...	СТ42	3
	Леухин П.Г.	МО51	4
	Николаева ...	БУ53	5
	Иваненко	ВВ2	1
	Иваненко	ТТ	2
*	NULL	NULL	NULL

Результат выполнения операции DELETE представлен на рисунке ниже:

	ФИО	Группа	Курс
▶	Иванов А.И.	ММ11	2
	Петрова И....	ПИ21	2
	Сидорова ...	МО31	4
	Кожевнико...	БУ33	5
	Пальчиков...	ММ12	1
	Царегород...	ПИ41	2
	Баранова Г...	СТ42	3
	Леухин П.Г.	МО51	4
	Николаева ...	БУ53	5
	Иваненко	ВВ2	1
	Иваненко	ТТ	2
*	NULL	NULL	NULL

Удаление строк из таблицы, на которой основано представление, невозможно, если:

- ✓ предложение FROM в определении представления содержит более чем одну таблицу, и список столбцов содержит столбцы более чем из одной таблицы;
- ✓ столбец в представлении создается из агрегатной функции;
- ✓ инструкция SELECT представления содержит предложение GROUP BY или параметр DISTINCT.

2. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Дайте определение понятию «представление (виртуальная таблица)».
2. Назовите основное отличие базовых таблиц и представлений.

3. Назовите следующие команды: создание индексов, создание хранимых процедур, создание триггеров, создание представлений.
4. Назовите основные задачи, которые решает создание представлений.
5. Перечислите случаи, когда в представлении требуется обязательно явно указывать имена столбцов.
6. Назовите команду удаления представления и команду обновления представления.
7. Поясните, с какой целью применяется предложение WITH CHECK OPTION с инструкциями INSERT и UPDATE.
8. Перечислите случаи, когда нельзя выполнять инструкции INSERT, UPDATE и DELETE применительно к представлениям.

ДОМАШНЕЕ ЗАДАНИЕ

[1], страницы 301-319

ЛИТЕРАТУРА

1. Петкович, Д. Microsoft SQL Server 2012. Руководство для начинающих: пер. с английского / Д. Петкович. – СПб.: БХВ-Петербург, 2013. – 816 с.: ил.
2. Сеть разработчиков Microsoft [Электронный ресурс]. – Режим доступа: <https://msdn.microsoft.com/ru-ru/library>

Преподаватель

С.В. Банцевич

Рассмотрено на заседании цикловой
комиссии программного обеспечения
информационных технологий №10
Протокол № от « » _____ 2017
Председатель ЦК С.В. Банцевич