Программирование баз данных

Лабораторная работа 10
Создание представлений

Даты выполнения лабораторной работы:

ФИб-21, 1 п/г	ФИб-21, 2 п/г	ФИб-22
30.04.2020	30.04.2020	27.04.2020

Представление – это виртуальная таблица, содержимое которой определяется запросом. Как и реальная таблица, представление состоит из набора именованных столбцов и строк данных. Представление существует в БД в виде хранимого набора значений данных, только если оно проиндексировано. Используются строки и столбцы данных из таблиц, на которые выполняются ссылки в запросе, определяющем представление, и они создаются динамически при ссылке на представление. Таблицы, запрашиваемые в представлении, называются базовыми таблицами.

Типичные примеры представлений:

- подмножество строк (горизонтальное представление) или столбцов базовой таблицы (вертикальное представление);
- объединение одной или нескольких базовых таблиц;
- соединение одной или нескольких базовых таблиц;
- статистическая сводка базовой таблицы;
- подмножество другого представления или некоторая комбинация представлений и базовых таблиц.

Преставления обычно используются в следующих случаях:

- сокрытие от пользователя ненужных или конфиденциальных данных;
- упрощение работы с данными;
- повышение безопасности благодаря предоставлению пользователям возможности обращаться к данным через представление без получения разрешения на прямой доступ к основным базовым таблицам представления;
- обеспечение обратной совместимости путём определения эмуляции представлением таблицы, которая существовала ранее, но схема которой была изменена;

- определение набора данных, которые пользователь может экспортировать и импортировать в SQL Server;
- обеспечение консолидированного представления секционированных данных, т. е. сходных данных, которые хранятся в разных таблицах.

Синтаксис оператора создания представления в MS SQL Server.

```
CREATE VIEW [ schema_name. ] view_name [ (column [ ,...n ]) ]
[ WITH <view_attribute> [ ,...n ] ]
AS select_statement
[ WITH CHECK OPTION ] [ ; ]

<view_attribute> ::= {
  [ ENCRYPTION ]
  [ SCHEMABINDING ]
  [ VIEW_METADATA ]
}
```

Аргументы

schema_name – имя схемы, которой принадлежит представление.

view_name – имя представления (должно соответствовать требованиям, предъявляемым к идентификаторам).

column – имя, которое будет иметь столбец в представлении. Имя столбца требуется только в тех случаях, когда столбец формируется на основе арифметического выражения, функции или константы, если два или более столбцов могут по иной причине получить одинаковые имена или если столбцу представления назначается имя, отличное от имени столбца, от которого он произведён. Назначать столбцам имена можно также в инструкции SELECT. Если аргумент **column** не указан, столбцам представления назначаются такие же имена, которые имеют столбцы в инструкции SELECT.

AS – определяет действия, которые должны быть выполнены в представлении.

select_statement – инструкция SELECT, которая определяет представление. В этой инструкции можно указывать более одной таблицы и другие представления. Представление не обязательно является простым подмножеством строк и столбцов одной конкретной таблицы. С помощью

предложения SELECT можно создавать представление, использующее более одной таблицы, или другие представления любой степени сложности.

Предложения SELECT, используемые в определении представления, не могут включать следующие элементы:

- предложение ORDER BY, если только в списке выбора инструкции SELECT нет также предложения TOP;
- ключевое слово INTO;
- предложение OPTION;
- ссылку на временную таблицу или табличную переменную.

В аргументе **select_statement** можно использовать функции и множественные инструкции SELECT, разделённые оператором UNION или UNION ALL.

CHECK OPTION – обеспечивает соответствие всех выполняемых для представления инструкций модификации данных критериям, заданным при помощи аргумента **select_statement**. Если строка изменяется посредством представления, предложение **WITH CHECK OPTION** гарантирует, что после фиксации изменений доступ к данным из представления сохранится.

ENCRYPTION – выполняет шифрование элементов представления sys.syscomments, содержащего текст инструкции CREATE VIEW. Использование предложения **WITH ENCRYPTION** предотвращает публикацию представления в рамках репликации SQL Server.

SCHEMABINDING – привязывает представление к схеме базовой таблицы или таблиц. Если аргумент **SCHEMABINDING** указан, нельзя изменить базовую таблицу или таблицы таким способом, который может повлиять на определение представления. При использовании аргумента **SCHEMABINDING** инструкция **select_statement** должна включать двухкомпонентные (schema.object) имена таблиц, представлений или пользовательских функций, упоминаемых в предложении. Все указанные в инструкции объекты должны находиться в одной БД.

VIEW_METADATA – указывает, что экземпляр SQL Server возвратит в API-интерфейсы DB-Library, ODBC и OLE DB сведения метаданных о представлении вместо базовой таблицы или таблиц, когда метаданные режима обзора затребованы для запроса, который ссылается на представление. Метаданные режима обзора — это дополнительные метаданные, которые экземпляр SQL Server возвращает вышеназванным клиентским API-интерфейсам. Эти метаданные позволяют клиентским API-интерфейсам реализовывать обновляемые клиентские курсоры. Метаданные режима обзора

содержат сведения о базовой таблице, которой принадлежат столбцы в результирующем наборе. Для представлений, созданных с применением предложения **VIEW_METADATA**, метаданные режима обзора возвращают имя представления, а не имена базовых таблиц при описании столбцов из представления в результирующем наборе.

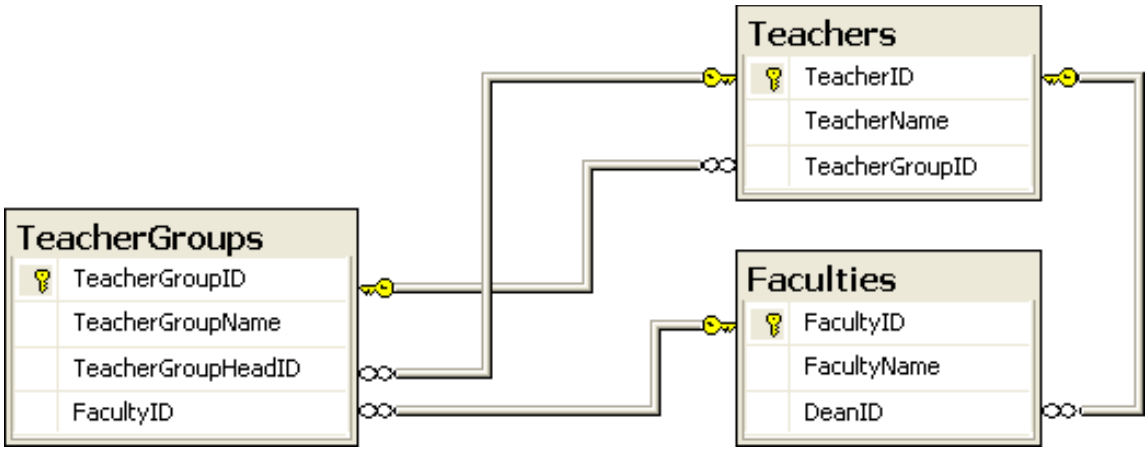
Для представлений определены операторы ALTER VIEW и DROP VIEW.

Пример. Использование простой инструкции CREATE VIEW.

```
USE AdventureWorks2008;
GO
IF OBJECT_ID ('hiredate_view', 'V') IS NOT NULL
DROP VIEW hiredate_view;
GO
CREATE VIEW hiredate_view
AS
SELECT c.FirstName, c.LastName, c.BusinessEntityID, e.HireDate
FROM HumanResources.Employee e JOIN Person.Person c
     ON e.BusinessEntityID = c.BusinessEntityID;
GO
```

Для выполнения заданий 1-4 нужно использовать БД, созданную при выполнении лабораторной работы 4.

Для выполнения задания 5 – БД, созданную на лабораторных работах 1-3.



Teachers		
TeacherID	**TeacherName**	**TeacherGroupID**
1	Лавров С.Д.	3
2	Соболева О.К.	1
3	Арасланова М.Л.	1
4	Коршунова Е.М.	2
5	Петухова А.Н.	2

TeacherGroups			
TeacherGroupID	**TeacherGroupName**	**TeacherGroupHeadID**	**FacultyID**
1	Кафедра ТМ	2	1
2	Кафедра ИР	5	2
3	Кафедра ФСП	1	2

Faculties		
FacultyID	**FacultyName**	**DeanID**
1	Факультет МК	3
2	Факультет ВТиТ	4

Задание 1. Создать представление, выводящее данные о каждой кафедре с указанием названия кафедры, заведующего кафедрой и декана факультета, которому принадлежит кафедра.

Задание 2. Создать запрос, возвращающий данные о кафедрах из созданного в задании 1 представления.

Задание 3. Создать представление, выводящее данные о каждом сотруднике с указанием ФИО, названия кафедры и названия факультета, на котором работает сотрудник. В представление включить только сотрудников факультета ВТиТ.

Задание 4. Создать запрос, возвращающий данные о сотрудниках из созданного в задании 3 представления.

Задание 5. Создать несколько представлений (не менее двух) в БД по варианту. Обосновать свой выбор. Выяснить, какие из представлений являются обновляемыми, а какие нет. Почему?