

## Лабораторная работа № 4

Тема: " Запросы на выборку данных по вариантам индивидуальных заданий "

**Цель работы:** усвоить способы создания представлений средствами СУБД MS SQL Server.

### Краткие теоретические сведения

#### Представление

Представление - это виртуальная таблица, определяемая запросом, содержащим оператор SELECT. Эта виртуальная таблица состоит из данных одной или нескольких реальных таблиц, а для пользователей представление выглядит, как реальная таблица. С представлением можно работать, как с обычной таблицей, ним можно применять операции SELECT, INSERT, UPDATE и DELETE. Можно создавать несколько типов представлений, каждый из которых имеет свои преимущества в определенных ситуациях.

- Подмножество колонок или строк таблицы. Это наиболее распространенный тип представления, который можно применять для упрощения кажущейся сложности базы данных для конечных пользователей.
- Связывание двух и более таблиц. Сложные операции связывания можно упростить, если использовать для этого представление.
- Агрегированная информация. Можно создать представление, содержащее агрегированные данные. Этот тип представления также используется для упрощения сложных операций.
- Способ внесения изменений в данные

Хотя с точки зрения функциональных возможностей представления аналогичны таблицам, тем не менее, при использовании операций модификации данных с помощью представлений необходимо учитывать следующие особенности:

- Если оператор выборки, лежащий в основе представления содержит операцию соединения (то есть представление создано на основе нескольких таблиц), то с помощью этого представления как правило, невозможно вставлять или удалять данные;
- Если представление ссылается только на одну таблицу, то вставка данных с помощью оператора INSERT с использованием представления может осуществляться, но при условии, что в представлении обеспечивается доступ ко всем обязательным (не допускающим неопределенных значений) столбцам таблицы или для этих столбцов заданы значения, применяемые по умолчанию.

Представления можно также использовать для объединения секционированных данных. Данные большой таблицы можно секционировать на несколько меньших таблиц, чтобы облегчить управление этими данными, а затем с целью упрощения доступа можно использовать представления для слияния этих таблиц в одну более крупную виртуальную таблицу.

Одним из преимуществ использования представлений является то, что они всегда содержат самые свежие данные. Оператор SELECT, определяющий представление, выполняется только при доступе к этому представлению, поэтому все изменения, внесенные в базовые таблицы представления, отражаются в этом представлении.

Еще одним преимуществом использования представлений является то, что представление может иметь уровень безопасности, отличный от его базовой таблицы. Запрос, определяющий представление, запускается с уровнем безопасности пользователя, создавшего это представление. Можно использовать представление, чтобы маскировать (скрыть) данные, которые вы не хотите показывать определенным классам пользователей.

SQL Server налагает несколько ограничений на создание и использование представлений. Это следующие ограничения:

- Ограничения по колонкам. Представление может использовать до 1024 колонок таблицы.
- Ограничение базы данных. Представление можно создать по таблице только в той базе данных, к которой осуществляет доступ создатель представления.
- Ограничение безопасности. Создатель представления должен иметь доступ ко всем колонкам, входящим в это представление.
- Правила целостности данных. Любые обновления, модификации и т.п., вносимые в представление, не могут нарушать правил целостности данных. Например, если базовая таблица не допускает null-значений, то они также не допускаются этим представлением.

- Ограничение на количество уровней вложенности представлений. Представления могут формироваться на основе других представлений. Допускается до 32 уровней вложенности представлений.
- Ограничение оператора SELECT. Используемый для представления оператор SELECT не может содержать оператора ORDER BY, COMPUTE или COMPUTE BY или ключевого слова INTO.

Представление можно создать тремя способами:

- с помощью оператора T-SQL CREATE VIEW.
- с помощью SQL Server Management studio

#### Использование T-SQL для создания представления

Оператор CREATE VIEW имеет следующий синтаксис:

CREATE VIEW имя представления [(колонка, колонка...)]

[WITH ENCRYPTION]

AS

ваш оператор SELECT

[WITH CHECK OPTION]

Создавая представление, вы можете активизировать два средства, которые изменяют поведение представления. Для активизирования этих средств нужно включить в оператор T-SQL ключевые слова WITH ENCRYPTION и/или WITH CHECK OPTION.

Ключевое слово WITH ENCRYPTION указывает, что определение представления (оператор SELECT, определяющий представление) должно шифроваться.

Ключевое слово WITH CHECK OPTION указывает, что операции модифицирования данных, применяемые к представлению, должны отвечать критериям, содержащимся в операторе SELECT. Например, можно запретить операцию модифицирования данных, применяемую к представлению для создания строки таблицы, которая не видна внутри этого представления. Предположим, что определяется представление для выборки информации обо всех служащих финансового отдела (finAnce department). Если ключевое слово WITH CHECK OPTION не включено в оператор, то вы можете изменить значение finAnce колонки department на значение, указывающее другой отдел. Но если это ключевое слово указано, то данное изменение не будет допускаться, поскольку изменение значения колонки department в какой-либо строке сделает эту строку недоступной из данного представления. Ключевое слово WITH CHECK OPTION указывает, что вы не можете сделать какую-либо строку недоступной из представления, внося какое-либо изменение внутри этого представления.

Оператор SELECT можно изменять для создания любого нужного вам представления. Его можно использовать для выборки подмножества колонок или подмножества строк либо для выполнения какой-либо операции связывания (join).

Представление, содержащее подмножество колонок, может оказаться полезным, если вам требуется обеспечить безопасность таблицы, которая должна быть доступна пользователям лишь частично. Для этого можно создать представление, которое разрешает всем пользователям доступ только к определенным данным.

Чтобы создать представление по таблице Employee, в котором имеется доступ только к колонкам name (имя), phone (телефон) и office (комната), используйте следующий оператор T-SQL:

```
CREATE VIEW emp_vw
```

```
AS
```

```
SELECT name, phone, office
```

```
FROM Employee
```

Результирующее представление будет содержать колонки. Хотя эти колонки также существуют в базовой таблице, пользователи, имеющие доступ к данным через это представление, могут видеть эти колонки только в этом представлении. А поскольку представление может иметь уровень безопасности, отличный от базовой таблицы представления, это представление можно предоставлять для доступа любому пользователю, в то время как образующая таблица останется защищенной.

Представление, состоящее из подмножества строк, можно использовать для ограничения доступа путем селекции строк, доступных для пользователей. В этом примере вместо ограничения колонок мы ограничим строки, задав их в предложении WHERE, как это показано ниже:

```
CREATE VIEW emp_vw2 AS
```

```
SELECT *
```

```
FROM Employee
```

WHERE Dept = 1

Результирующее представление будет содержать только строки, которые удовлетворяют условию. Представлению с подмножеством строк, как и представлению с подмножеством колонок можно присвоить уровень безопасности, отличный от базовой таблицы или таблиц представления.

Определяя связывание в представлении, вы можете упрощать операторы T-SQL, используемые для доступа к данным, по сравнению с операторами, содержащими оператор JOIN. Рассмотрим пример. Предположим, у нас имеются две таблицы - MAnagei-и Employee2.

Следующий оператор выполняет связывание таблиц Employee2 и MAnager в одну виртуальную таблицу:

```
CREATE VIEW org_chart
AS
SELECT Employee.ename, MAnager.mname
FROM Employee, MAnager
WHERE Employee.mAnagend = mAnager.id
GROUP BY MAnager.mname, Employee.ename
```

В данном примере указанные две таблицы связаны значением mAnager\_id. Результирующие данные, содержащиеся в представлении org\_chart, сгруппированы по имени руководителя (mAnager\_name). Отметим, что если руководитель указан в таблице MAnager, но в таблице Employee2 нет ни одного служащего (employee) для этого руководителя, то в представлении нет ни одной записи для этого руководителя. В представлении также нет ни одной записи для служащего, содержащегося в таблице Employee2, но не имеющего соответствующего руководителя в таблице MAnager. Все это видят пользователи в виртуальной таблице со служащими и руководителями.

Агрегированные представления могут оказаться полезными во многих случаях, таких как считывание средних значений и сумм по отделам. Следующий оператор задает представление, в котором используется функция агрегирования SUM по таблице Employee:

```
CREATE VIEW sal_vw
AS
SELECT dept, SUM(Salary)
FROM Employee
GROUP BY dept
```

Представления можно использовать для слияния секционированных данных в одну виртуальную таблицу. Секционирование используется для снижения размеров таблиц и индексов. Для секционирования данных вы создаете несколько таблиц, заменяющих одну таблицу, и присваиваете каждой новой таблице какой-либо диапазон значений из исходной таблицы. Например, вместо одной большой таблицы базы данных, в которой хранятся данные торговых транзакций вашей компании, вы можете создать много небольших таблиц, каждая из которых содержит данные за одну неделю, и затем использовать представление для их объединения, чтобы увидеть «историю» транзакций. Представление выглядит для пользователей как одна большая таблица, но она образована многими таблицами, каждая из которых имеет собственный индекс.

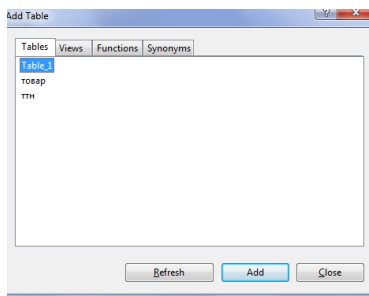
Чтобы создать представление, в котором объединяются секционированные данные, вы должны сначала создать секционированные таблицы. В каждой таблице будут храниться данные за определенный период - обычно за неделю или месяц. Создав эти таблицы, вы можете использовать оператор UNION ALL, чтобы создать представление, содержащее все данные. Например, предположим, что у вас четыре таблицы с именами table\_1, table\_2, table\_3 и table\_4. Следующий оператор создает одну большую виртуальную таблицу, содержащую все данные из этих таблиц:

```
CREATE VIEW partview
AS
SELECT * FROM table_1
UNION ALL
SELECT * FROM table_2
UNION ALL
SELECT * FROM table_3
UNION ALL
SELECT * FROM table_4
```

#### Использование Management Studio создания представления

1. В окне Management Studio раскройте папку Databases (Базы данных) для сервера, на котором находится ваша база данных.

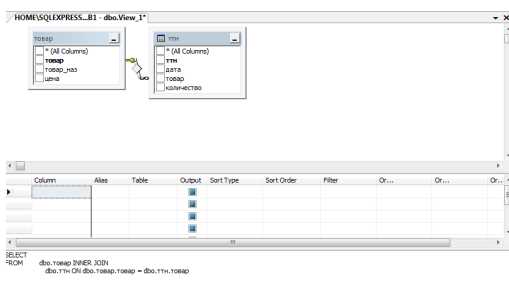
2. Раскройте папку с вашей БД и найдите в ней папку Views. Щелкните на ней правой кнопкой мыши. Выберите в появившемся контекстном меню команду New View (Создать представление), чтобы открыть диалоговое окно Add Table.



Это диалоговое окно позволяет выбирать таблицы, данные которых должны быть включены в представление. Если нужно выбрать для представления больше одной таблицы нужно удерживать нажатой клавишу <Ctrl>, выделяя все необходимые таблицы.

В этом диалоговом окне есть и другие вкладки: Views, Functions, Synonyms, находящиеся в его верхней части. В представлении могут быть предусмотрены непосредственные ссылки на включенные в него объекты, и данное окно представляет возможность сразу же ввести необходимые ссылки в определение представления.

После этого нажмите кнопку Add. Откроется окно редактора представлений.



Это окно состоит из четырех подокон, которые можно разворачивать и сворачивать независимо друг от друга:

- **Панель схемы (Diagram).** Показывает данные таблицы, которые используются для создания представления. В этой панели можно выбирать колонки, нажав правой кнопкой мыши на выбранной колонке данных и выбрав команду Add to Select в контекстном меню.
- **Панель-сетка Grid (Criteria).** Показывает колонки, выбранные из таблицы или базовых таблиц. В этой панели можно выбирать колонки. Колонки Sort Type и Sort Order позволяют задать необходимость сортировки и порядок сортировки по данному полю представления соответственно. В поле Criteria можно указать условие, которому должно удовлетворять данное поле представления (добавляется к результирующему SQL-запросу с помощью AND-объединения), а в полях Or... определить условия на поле, добавляемые к результирующему SQL-запросу с помощью OR-объединения.
- **Панель SQL.** Показывает оператор SQL, используемый для определения данного представления. SQL Server генерирует этот оператор SQL, когда вы перетаскиваете элементы панели схемы и выбираете колонки в панели-сетке.
- **Панель результатов (Results).** Показывает строки, считанные из представления. Эта информация позволяет вам понять, как выглядят данные.

Вы можете задавать визуализацию этих панелей, щелкая на соответствующих кнопках в панели инструментов окна New View. Другие кнопки панели инструментов используются для некоторых важных функций. В следующем списке дается описание этих кнопок, начиная с левого края панели инструментов:

- **Save (Сохранить).** Сохраняет данное представление.
- **Show/Hide (Показать/Скрыть панели)** - четыре кнопки. Позволяют вам показывать или скрывать четыре панели окна New View.

- Execute (Выполнить). Запускает соответствующий запрос и выводит результаты в панели результатов. Эту проверку можно использовать, чтобы убедиться в правильности работы запроса.
- Cancel Execution And Clear Results (Прекратить выполнение и удалить результаты). Очищает панель результатов.
- Verify SQL (Проверить SQL). Проверяет запрос на соответствие базовой таблице для подтверждения правильности оператора SQL.
- Use GROUP BY (Использовать GROUP BY). Добавляет предложение GROUP BY к оператору в панели SQL.
- Add Table (Добавить таблицу). Позволяет добавить какую-либо таблицу к запросу.

3. Модифицируйте оператор SELECT в панели SQL в соответствии с оператором SELECT. Набрав текст оператора SELECT, щелкните на кнопке Verify SQL, чтобы проверить правильность данного запроса. Если это так, то вы должны щелкнуть на кнопке ОК.

4. Закончив проверку того, что представление отвечает вашим требованиям (с помощью панели результатов), и внося необходимые изменения, закройте окно New View. Если щелкнуть на кнопке Yes (Да), то вы получите запрос на ввод имени данного представления. Введите описательное имя вашего представления и сохраните представление, щелкнув на кнопке ОК.

При нажатии кнопки F5 отображается таблица, в которой показывается содержимое сформированного представления, подобное отображению содержимого простой таблицы. Сохраняется сформированное представление данных с помощью нажатия на File > Save.

Ваше представление теперь доступно для использования. Вы можете использовать Management Studio, чтобы задать свойства нового представления, включая полномочия доступа.

#### Использование Management Studio для изменения и удаления представлений

Для изменения и удаления представлений с помощью Management Studio выполните следующие шаги.

1. В окне Management Studio раскройте папку Databases на нужном сервере, раскройте базу данных, содержащую представление, которое хотите удалить или изменить, и затем щелкните на Views (Представления), чтобы в правой панели этого окна появился список представлений.
2. Щелкните правой кнопкой мыши на имени представления, которое хотите модифицировать или удалить. Появится контекстное меню. Чтобы удалить представление, выберите в этом меню пункт Delete (Удалить). Чтобы изменить представление, выберите пункт Design View (Разработка представления).
3. Если выбран пункт Delete, то появится диалоговое окно Drop Objects (Удаление объектов). Щелкните на кнопке Show Dependencies (Показать зависимости), чтобы увидеть базовую структуру представления. Здесь вы можете видеть, от каких таблиц зависит представление. Если выбрано представление типа join (связывание) или union (объединение), то вы увидите все участвующие таблицы; если это представление по определенным колонкам или строкам, то вы увидите только одну таблицу. Когда вы будете готовы удалить выбранное представление, щелкните на кнопке Drop All (Удалить все) в диалоговом окне Drop Objects.
4. Если в контекстном меню выбран пункт Design View, то появится окно Design View. Вы можете использовать окно Design View для модифицирования вашего представления таким же способом, что и при создании представления в окне New View. После внесения необходимых изменений в представление закройте окно Design View, щелкнув на кнопке Close (Заккрыть) этого окна. Затем вы получите предложение сохранить это представление.

Закончив модифицирование представления, вы можете задать полномочия доступа к этому представлению. Сначала откройте окно View Properties, щелкнув правой кнопкой мыши на имени представления в окне Management Studio выбрав из контекстного меню пункт Properties. Затем щелкните на Permissions (Полномочия), чтобы вывести на экран полномочия доступа для данного представления.

#### Использование T-SQL для изменения и удаления представлений

Для изменения представлений с помощью T-SQL используйте оператор ALTER VIEW. Оператор ALTER VIEW аналогичен оператору CREATE VIEW и имеет следующий синтаксис:

```
ALTER VIEW имя_представления [(колонка, колонка,...)]
[WITH ENCRYPTION]
AS
ваш оператор SELECT
[WITH CHECK OPTION]
```

Единственным отличием между операторами ALTER VIEW и CREATE VIEW является то, что оператор CREATE VIEW не будет выполняться, если представление уже существует, а оператор ALTER VIEW не будет выполняться, если указанное представление не существует.

Для удаления представления используйте оператор DROP VIEW. Оператор DROP VIEW имеет следующий простой синтаксис:

DROP VIEW *имя\_представления*

Для получения информации о представлении используется хранимая процедура: **sp\_help** *ИмяПредставления*, которая отображает список полей представления с описанием их свойств. Для отображения кода, с помощью которого создано представление, можно воспользоваться хранимой процедурой: **sp\_helptext** *ИмяПредставления*.

Список объектов, от которых зависит представление, может быть получен выполнением хранимой процедуры: **sp\_depends** *ИмяПредставления*.

**Лабораторная работа** рассчитана на 2 часа аудиторных занятий и состоит в изучении теоретического материала и получении практических навыков по созданию представлений базы данных. Сдача лабораторной работы заключается в ответах на контрольные вопросы и демонстрации индивидуального задания.

#### **Содержание отчета (в электронном виде):**

- 1.Название и цель работы
- 2.Индивидуальное задание
- 3.Скрипты на создание представлений

#### **Индивидуальное задание**

1. Создать представления для реализации отчетов в соответствии с требованиями индивидуального варианта задания на разработку курсового проекта
2. Создать представление по своему усмотрению на основании двух или более таблиц вашей базы данных с целью просмотра и корректировки базовых таблиц
3. Создать представление по своему усмотрению на основании одной таблицы вашей базы данных с целью корректировки базовой таблицы

#### **Контрольные вопросы**

- 1.Назначение механизма представлений
- 2.Классификация представлений
- 3.Способы создания представлений