**Представления View**

Представление (view) в Microsoft SQL Server представляет собой виртуальную таблицу, которая отображает данные из одной или нескольких таблиц в базе данных. Она не хранит данные физически, а предоставляет доступ к данным из других таблиц и представлений. Представления могут быть использованы в запросах, как и обычные таблицы, и могут содержать любые SELECT-операторы, включая объединения и агрегатные функции.

Преимущества использования представлений включают:

Упрощение запросов к базе данных: представления могут абстрагировать сложные запросы к базе данных, сделав их более понятными и легкими в использовании.

Обеспечение безопасности: представления могут использоваться для скрытия некоторых данных из таблицы и предоставления только определенных данных пользователям, которые должны иметь к ним доступ.

Уменьшение дублирования кода: представления могут использоваться для повторного использования запросов в нескольких местах в базе данных.

Создание представлений осуществляется с помощью оператора CREATE VIEW. Например, следующий запрос создает представление, которое отображает все заказы, которые были сделаны в последние 30 дней:

CREATE VIEW RecentOrders

AS

SELECT OrderID, OrderDate, CustomerID, TotalAmount

FROM Orders

WHERE OrderDate >= DATEADD(day, -30, GETDATE())

После создания представления его можно использовать как обычную таблицу в запросах к базе данных:

SELECT \*

FROM RecentOrders

WHERE CustomerID = 'ALFKI'

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.

Drop view Название – удаление представления

Оператор SQL DISTINCT используется для указания на то, что следует работать только с уникальными значениями столбца.

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

− запрос не должен содержать секцию группировки GROUP BY;

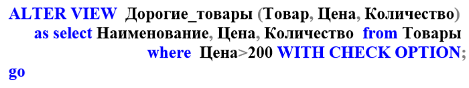
− запрос не должен применять агрегатные функции, опции DISTINCT и TOP, операторы UNION, INTERSECT и EXCEPT;

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

− в секции FROM запроса должна указываться только одна таблица.

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

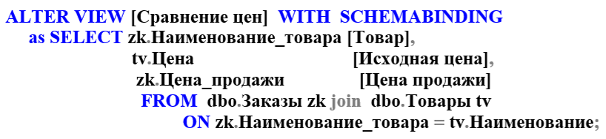
Например, можно изменить представление **Дорогие\_товары**:



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

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

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).

Опция SCHEMABINDING в представлениях (views) в Microsoft SQL Server указывает, что представление зависит от схемы базы данных и от объектов, используемых в нем, и предотвращает изменения этих объектов, которые могут нарушить работу представления.

Когда опция SCHEMABINDING используется при создании представления, это означает, что представление закрепляет структуру и имена объектов в других таблицах и представлениях, используемых в его определении. В результате, любые изменения в этих объектах, таких как удаление или изменение имени столбца, которые могут нарушить ссылки в представлении, будут недопустимыми.

Использование опции SCHEMABINDING может привести к некоторым преимуществам, таким как повышение производительности и уменьшение времени выполнения запросов. При использовании опции SCHEMABINDING в представлениях SQL Server может избежать выполнения дополнительных проверок целостности данных, поскольку она знает, что определенные свойства таблиц не могут изменяться.

Однако, использование опции SCHEMABINDING может стать проблемой, если вам нужно изменить определение объектов, на которые ссылается представление. При использовании опции SCHEMABINDING вы не сможете выполнить такие изменения до тех пор, пока не удалите представление.

Пример создания представления с опцией SCHEMABINDING:

CREATE VIEW dbo.myView

WITH SCHEMABINDING

AS

SELECT col1, col2, col3

FROM dbo.myTable

WHERE col1 = 'SomeValue'

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 schema.db\_object, где schema - 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.

Оператор PIVOT в Microsoft SQL Server используется для преобразования строковых данных в столбцы. Он используется для агрегации и транспонирования данных из столбцов таблицы в новые столбцы в соответствии с определенными условиями.

Оператор PIVOT включает следующие элементы:

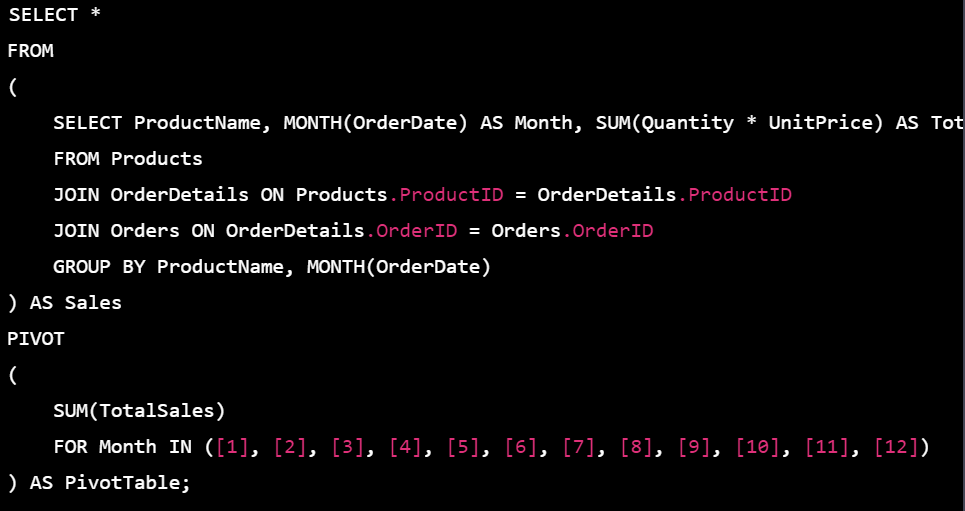
- Определение агрегатной функции: в первую очередь нужно определить, какие данные вы хотите агрегировать. Например, вы можете использовать функции SUM(), COUNT() или AVG() для агрегирования числовых данных.

- Определение столбцов для транспонирования: следующим шагом является определение столбцов, которые нужно транспонировать. Эти столбцы будут преобразованы в новые столбцы в выходной таблице.

- Определение столбцов для группировки: затем нужно определить столбцы, по которым нужно группировать данные.

- Использование ключевого слова PIVOT: наконец, нужно использовать ключевое слово PIVOT, чтобы выполнить транспонирование данных.

Пример использования оператора PIVOT для агрегирования продаж по месяцам и продуктам:



В этом запросе мы используем оператор PIVOT, чтобы преобразовать строки в столбцы. Сначала мы используем подзапрос для агрегирования продаж по месяцам и продуктам. Затем мы используем оператор PIVOT, чтобы преобразовать строки с месяцами в столбцы. Функция SUM() используется для агрегирования данных, а ключевое слово FOR используется для определения списка столбцов, которые нужно транспонировать.