

Использование каталога представлений для просмотра информации по объектам SQL-сервера

Контекст	СУБД Microsoft SQL Server 2005 (9), далее MSSS
Целевая аудитория	Разработчики клиент-серверных приложений на платформе MSSS; люди, увлеченные системами управления базами данных и технологиями Microsoft
Ключевые слова для поиска дополнительной информации в библиотеке MSDN	catalog views, system tables

Предположим, что у нас в распоряжении имеется рабочая станция. На нее установлено стандартное клиентское программное обеспечение для управления серверами баз данных MSSS. На этой рабочей станции зарегистрирован один экземпляр SQL-сервера (далее, сервер), расположенного где-то в просторах нашей локальной сети. На этом сервере мы ведем разработку наших приложений.

Стандартным средством для управления сервером является Microsoft SQL Server Management Studio (SSMS) (Студия для управления SQL-сервером). Под управлением понимается выполнение команд манипулирования различными объектами сервера: регистрационными именами, базами данных, таблицами, хранимыми процедурами, пользовательскими функциями и другими объектами.

Для просмотра информации по объектам сервера SSMS предоставляет разработчику два средства:

- Обозреватель объектов (Object Explorer);
- Страница "Резюме" (Summary Page).

Обозреватель объектов представляет все объекты сервера в виде иерархического дерева. Каждый узел дерева — объект определенного типа: пользователь, база данных, таблица, представление и т. п. Тип объекта определяет множество доступных свойств и это очевидно, ведь свойства таблицы, отличаются от свойств регистрационного имени.

Страница Резюме позволяет перемещаться по серверу как по файловой системе. Здесь все представляется в виде иерархии папок. Каждая папка может содержать как объекты сервера (файлы, т. е. целостные сущности), так и другие вложенные папки.

Как показывает практика, оба эти средства не могут удовлетворить наши потребности, так как они не обеспечивают нас достаточной функциональностью, удобством и скоростью отклика на действия.

Существует третье средство для просмотра служебной информации по объектам сервера — это Catalog view или Каталог представлений (далее, Каталог). Каталог — это множество представлений, содержащих метаданные всех объектов сервера. Все такие представления принадлежат схеме sys, то есть начинаются с префикса sys. Представления имеют очень простые интуитивные наименования. Например, представление с метаданными таблиц называется sys.tables (таблицы), хранимых процедур — sys.procedures (процедуры), индексов — sys.indexes (индексы), ограничений внешнего ключа — sys.foreign_keys (внешние ключи) и т. д.

Чтобы получить информацию через Каталог , следует написать и выполнить простое SQL-утверждение, например:

-- получение основной информации по всем таблицам базы данных;

```
SELECT
    SCHEMA_NAME( schema_id ) 'Схема',
    name 'Наименование',
    create_date 'Дата создания',
    modify_date 'Дата изменения'
FROM sys.tables
```

-- получение информации по параметрам хранимой процедуры; имя нужной нам процедуры определяется через переменную @procedure_name;

```
DECLARE @procedure_name sysname
SET @procedure_name = 'first_select' -- имя таблицы;

SELECT * FROM sys.parameters
    WHERE object_id IN (
        SELECT
            object_id
            FROM sys.procedures PRC
            WHERE PRC.name LIKE @procedure_name
    )
```

-- Получение списка родительских таблиц;

```
SELECT [name] FROM sys.tables
    WHERE object_id IN(
        SELECT referenced_object_id
            FROM sys.foreign_keys
    )
```

-- Получение информации обо всех дочерних таблицах, для которых разрешено каскадное удаление записей;

```
SELECT * FROM sys.tables
    WHERE object_id IN (
        SELECT parent_object_id
        FROM sys.foreign_keys
        WHERE delete_referential_action = 1 -- 1 означает каскадное
        удаление;
    )
```

-- Получение информации по всем колонкам таблицы, позволяющим содержать нуль-значения; имя нужной нам таблицы определяется через переменную @table_name;

```
DECLARE @table_name sysname
SET @table_name = 'first' -- имя таблицы;
SELECT T.[name], C.* FROM sys.tables T
    JOIN sys.columns C ON C.object_id = T.object_id
    WHERE C.is_nullable = 1 -- 1 означает, что колонка позволяет содержать нуль-значения;
    AND T.[name] LIKE @table_name
```

Разработка шаблонов запросов к Каталогу представлений

Помним, что сервер может содержать произвольное количество различных баз данных. Но из примеров мы видим, что , запросы не привязаны к какой либо конкретной базе данных. Это означает, что мы можем использовать данные запросы в любой базе данных без изменения кода шаблонов. В этом случае удобно добавить приведенные примеры в Обзорщик шаблонов (Template Explorer) и выработать у себя привычку обращаться к

этим шаблонам при всякой потребности в служебной информации по объектам базы данных.

Конечно, те шаблоны, в которых используются переменные, то есть шаблоны, содержащие фрагмент "DECLARE @" , для удобства использования могут быть сделаны параметризуемыми.

Давайте рассмотрим как это делается на конкретном примере. Пусть в нашей базе данных определены две схемы work (работа), home (дом), каждая из которых содержит свое множество объектов базы данных. Первая схема содержит объекты, относящиеся к работе, вторая — содержит объекты, относящиеся к дому.

Следующий запрос возвращает информацию по всем объектам, принадлежащим схеме home:

-- Получение всех объектов базы данных, входящих в схему home:

```
DECLARE @schema_name sysname
SET @schema_name = 'home'
SELECT * FROM sys.objects WHERE schema_id = SCHEMA_ID( @schema_name )
```

Для того, чтобы получить информацию по всем объектам, принадлежащим схеме work, можно использовать следующий запрос:

-- Получение всех объектов базы данных, входящих в схему work:

```
DECLARE @schema_name sysname
SET @schema_name = 'work'
SELECT * FROM sys.objects WHERE schema_id = SCHEMA_ID( @schema_name )
```

Как мы видим, эти запросы отличаются лишь одним словом, значит этот запрос можно преобразовать в шаблон с одним параметром. Параметром будет изменяемая часть запроса, в нашем примере — наименование схемы.

Шаблон будет выглядеть следующим образом:

-- Шаблон для получения информации по объектам, принадлежащим интересующей нас схеме

```
DECLARE @schema_name sysname
SET @schema_name = '<schema_name, sysname, work>'
SELECT * FROM sys.objects WHERE schema_id = SCHEMA_ID( @schema_name )
```

Параметр шаблона состоит из трех частей: наименования, типа и значения по умолчанию. Для того, чтобы заменить параметр конкретным значением, мы можем использовать сочетание клавиш Ctrl + M или команду меню Query (Запрос) — Specify Values for Template Parameters (Задать значения для параметров шаблона). (Всю работу мы выполняем в редакторе сценариев приложения MSSS).

Представление sys.sql_modules

Одним из самых интересных представлений в Каталоге, с точки зрения автора, является представление sys.sql_modules. Это представление содержит колонку definition (определение), в ячейках которой содержится исходный код всех программ базы данных, а именно хранимых процедур, функций, триггеров, а также исходный код значений по умолчанию, представлений и правил.

С помощью представления sys.sql_modules можно производить поиск по исходному коду. Например, для того, чтобы получить список хранимых процедур, содержащих в своем исходном коде ключевое слово NOCOUNT, можно использовать следующий запрос:

-- Получение всех объектов, содержащих в своем исходном коде ключевое слово NOCOUNT;
SELECT [name] 'Наименование процедуры'

```

FROM sys.sql_modules SQLMD
JOIN sys.objects OBJ ON SQLMD.object_id = OBJ.object_id
WHERE definition LIKE '%NOCOUNT%'

```

Поле definition представления sys.sql_modules содержит информацию только по незашифрованным объектам, поэтому в результирующую выборку никогда не будут попадать объекты с зашифрованным исходным кодом, то есть объекты, определенные с опцией ENCRYPTION (шифровка). Для зашифрованных объектов, поле definition представления всегда содержит нуль-значение и это очевидно.

Другие полезные в практике запросы

Задача	Запрос
Получение списка таблиц, содержащих XML-документы	<pre> SELECT DISTINCT [name] FROM sys.tables WHERE object_id IN (SELECT object_id FROM sys.columns WHERE system_type_id IN (SELECT system_type_id FROM sys.types WHERE [name] LIKE 'xml') </pre>
Формирование сценария для назначения разрешений на выполнение всех хранимых процедур в текущей базе данных для пользователя с регистрационным именем maxim	<pre> SELECT 'GRANT EXECUTE ON ' + [name] + ' TO maxim' FROM sys.procedures </pre>