

# Системный каталог

**Ч**тобы осуществлять управление данными, СУБД должна отслеживать большое количество информации, определяющей структуру базы данных. В реляционной базе данных эта информация обычно хранится в *системном каталоге* — совокупности системных таблиц, используемых СУБД для собственных целей. Информация, хранящаяся в системном каталоге, описывает таблицы, представления, столбцы, привилегии и другие структурные элементы базы данных.

Хотя системный каталог предназначен, главным образом, для внутреннего применения, пользователи базы данных также могут получить доступ к системным таблицам с помощью стандартных запросов SQL. Таким образом, реляционная база представляет собой самоописываемую сущность; вы можете получить описание структуры базы данных, выполняя запросы к системным таблицам. Это используется в таких клиентских приложениях общего назначения, как модули формирования запросов и программы генерации отчетов, где для упрощения доступа к базе данных пользователям предоставляется на выбор список таблиц и столбцов.

В настоящей главе описываются системные каталоги нескольких популярных СУБД на базе SQL и содержащаяся в этих каталогах информация. Рассматривается также структура системного каталога в стандарте ANSI/ISO SQL.

## Что такое системный каталог

Системным каталогом называется совокупность специальных таблиц базы данных. Их создает, сопровождает и владеет ими сама СУБД. Эти *системные таблицы* содержат информацию, которая описывает структуру базы данных. Таблицы системного каталога создаются автоматически при создании базы данных. Обычно они объединяются под специальным системным идентификатором пользователя с таким именем, как SYSTEM, SYSIBM, MASTER или DBA.

При обработке инструкций SQL СУБД постоянно обращается к данным системного каталога. Например, чтобы обработать двухтабличную инструкцию SELECT, СУБД должна выполнить следующее:

- проверить, существуют ли две указанные таблицы;
- убедиться, что пользователь имеет разрешение на доступ к ним;
- проверить, существуют ли столбцы, на которые имеются ссылки в данном запросе;
- разрешить невалифицированные имена и установить, к каким таблицам они относятся;
- определить тип данных каждого столбца.

Так как информация о структуре базы данных хранится в системных таблицах, СУБД может использовать собственные методы и алгоритмы, чтобы быстро и эффективно извлекать информацию, необходимую для выполнения этих задач.

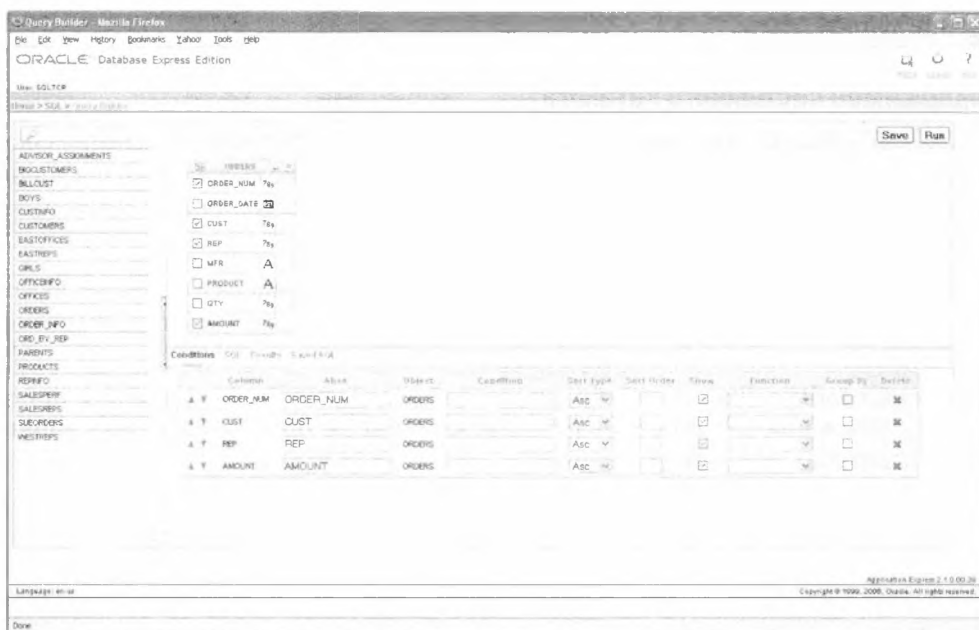
Если бы системные таблицы служили только для удовлетворения внутренних потребностей СУБД, то для пользователей базы данных они не представляли бы практически никакого интереса. Однако системные таблицы (или созданные на их основе представления), как правило, доступны также и для пользователей. Запросы к системным каталогам разрешены почти во всех базах данных для персональных и мини-компьютеров. В СУБД для мэйнфреймов или СУБД уровня предприятия такие запросы тоже допускаются, но администратор базы данных может ограничивать доступ к системному каталогу в качестве дополнительной меры обеспечения безопасности базы данных. С помощью запросов к системным каталогам вы можете получить информацию о структуре базы данных, даже если никогда раньше с ней не работали.

Пользователи могут только извлекать информацию из системного каталога. СУБД запрещает пользователям модифицировать системные таблицы непосредственно, так как это может нарушить целостность базы данных. СУБД сама вставляет, удаляет и обновляет строки системных таблиц во время модификации структуры базы данных. Изменения в системных таблицах происходят также в качестве побочного результата выполнения таких инструкций DDL, как CREATE, ALTER, DROP, GRANT и REVOKE. В некоторых СУБД даже инструкции DML, например INSERT и DELETE, могут модифицировать содержимое системных таблиц, отслеживая количество записей в таблицах.

## **Системный каталог и средства формирования запросов**

Одним из наиболее важных преимуществ системного каталога является то, что он позволяет создавать дружелюбные к пользователю программы формирования запросов, такие как Query Builder, являющаяся частью Oracle Application Express, поставляемой в составе Oracle Express Edition (рис. 16.1). Цель такой программы заключается в том, чтобы обеспечить простой и понятный способ доступа к базе данных пользователю, не знающему языка SQL. Обычно эта программа проводит пользователя через следующую последовательность действий.

1. Пользователь вводит свое имя и пароль, чтобы получить доступ к базе данных.
2. Программа формирования запросов отображает на экране список доступных таблиц.
3. Пользователь выбирает таблицу, после чего программа отображает на экране список столбцов данной таблицы.
4. Пользователь выбирает интересующие его столбцы — обычно при помощи щелчков мышью на именах столбцов, когда они появляются на экране.
5. Пользователь выбирает столбцы из других таблиц или ограничивает объем извлекаемых данных с помощью условия отбора.
6. Программа извлекает запрашиваемые данные и отображает их на экране.



**Рис. 16.1.** Oracle Query Builder — пример программы формирования запросов с дружественным интерфейсом

Программа формирования запросов общего назначения, подобная той, что представлена на рис. 16.1, будет применяться многими пользователями для доступа к самым различным базам данных. Ей неизвестна заранее структура базы данных в каждом конкретном случае, поэтому необходимо иметь возможность динамически получать информацию о таблицах и столбцах базы данных. Для достижения этой цели такая программа использует системный каталог.

## Системный каталог и стандарт ANSI/ISO

В стандарте SQL1 ничего не говорится о структуре и содержании системного каталога. Стандарт фактически не требует даже наличия самого системного ката-

лога. Однако во всех основных СУБД на базе SQL в той или иной форме он создается. Структура каталога и содержащиеся в нем таблицы значительно отличаются друг от друга в разных СУБД.

В связи с ростом популярности инструментальных программ общего назначения, предназначенных для работы с базами данных и требующих доступа к системному каталогу, в стандарт SQL (начиная с SQL2) включена спецификация набора представлений, обеспечивающая стандартизированный доступ к информации, которая обычно содержится в системном каталоге. СУБД, соответствующая стандарту SQL, должна поддерживать эти представления, все вместе именуемые INFORMATION\_SCHEMA. Так как эта схема сложнее, чем реальные системные каталоги, применяемые в большинстве коммерческих СУБД, и пока еще не пользуется широкой поддержкой, она рассматривается в отдельном разделе в конце данной главы.

## Содержимое системного каталога

Каждая таблица системного каталога содержит информацию об отдельном структурном элементе базы данных. В состав почти всех коммерческих реляционных СУБД входят, с небольшими различиями, системные таблицы, каждая из которых описывает один из следующих пяти элементов.

- **Таблицы.** В каталоге описывается каждая таблица базы данных: указывается ее имя, владелец, число содержащихся в ней столбцов, их размер и т.д.
- **Столбцы.** В каталоге описывается каждый столбец базы данных: приводится имя столбца, имя таблицы, которой он принадлежит, тип данных столбца, его размер, разрешены ли значения NULL и т.д.
- **Пользователи.** В каталоге описывается каждый зарегистрированный пользователь базы данных: указываются имя пользователя, его пароль в зашифрованном виде и другие данные.
- **Представления.** В каталоге описывается каждое представление, имеющееся в базе данных: указываются его имя, имя владельца, запрос, являющийся определением представления, и т.д.
- **Привилегии.** В каталоге описывается каждый набор привилегий, предоставляемых в базе данных: приводятся имена тех, кто предоставил привилегии, и тех, кому они предоставлены, указываются сами привилегии, объект, на которые они распространяются, и т.д.

В табл. 16.1 для основных реляционных СУБД даны имена системных таблиц, содержащих эту информацию. Некоторые типичные системные таблицы описываются более подробно далее; приводятся также примеры запросов к системному каталогу.

**Таблица 16.1. Избранные системные таблицы популярных СУБД**

СУБД	Таблицы	Столбцы	Пользователи	Представления	Привилегии
DB2 <sup>1</sup>	SCHEMATA	COLUMNS	DBAUTH	VIEWS	DBAUTH
	TABLES	KEYCOLUSE			SCHEMAAUTH
	REFERENCES	COLOPTIONS			TABAUTH
	TABOPTIONS				COLAUTH
	TABDEP				
Oracle <sup>2</sup>	CATALOG	TAB_COLUMNS	USERS	VIEWS	TAB_PRIVS
	OBJECTS	TAB_COLS			COL_PRIVS
	TABLES	LOBS			SYS_PRIVS
	SYNONYMS				
Informix	SYSTABLES	SYSCOLUMNS	SYSUSERS	SYSVIEWS	SYSTABAUTH
	SYSREFERENCES			SYSDEPEND	SYSCOLAUTH
	SYSYNONYMS				
Sybase	SYSDATABASES	SYSCOLUMNS	SYSUSERS	SYSOBJECTS	
	SYSOBJECTS			SYSCOMMENTS	
	SYSKEYS				
SQL Server <sup>3</sup>	DATABASES	COLUMNS	DATABASE_ PRINCIPALS	OBJECTS	DATABASE_ PERMISSIONS
	OBJECTS	FOREIGN_KEY_ COLUMNS	SQL_LOGINS	VIEWS	
	FOREIGN_KEYS	IDENTITY_ COLUMNS			
	REFERENCES				

<sup>1</sup> Таблицы DB2 используют квалификатор SYSCAT (например, SYSCAT.TABLES).

<sup>2</sup> Oracle предоставляет три версии многих представлений каталога с префиксами ALL\_, DBA\_ или USER\_ (например, ALL\_TABLES, DBA\_TABLES или USER\_TABLES). Версия ALL\_ показывает все объекты, к которым текущий пользователь имеет доступ, DBA\_ — все объекты во всей базе данных, а USER\_ — только объекты, владельцем которых является текущий пользователь.

<sup>3</sup> Представления каталога SQL Server имеют квалификатор SYS (например, SYS.DATABASES). Начиная с SQL Server 2000 происходит отказ от системных таблиц в пользу новых представлений каталога.

## Информация о таблицах

Во всех основных реляционных СУБД имеется системная таблица или представление, где отслеживается состояние всех таблиц базы данных. В DB2 это представление называется SYSCAT.TABLES. (Все системные представления в СУБД DB2 входят в схему, называющуюся SYSCAT, поэтому обладают именами вида SYSCAT.XXX)

В табл. 16.2 перечислены некоторые столбцы представления SYSCAT.TABLES. В нем для каждой таблицы, каждого представления или псевдонима, имеющих в базе данных, отводится одна строка. Примерно такая же информация предоставляется в соответствующих системных таблицах других СУБД.

**Таблица 16.2. Избранные столбцы таблицы SYSCAT.TABLES (СУБД DB2)**

Столбец	Тип данных	Информация
TABSCHEMA	VARCHAR (128)	Схема, содержащая таблицу, представление или псевдоним
TABNAME	VARCHAR (128)	Имя таблицы, представления или псевдонима
DEFINER	VARCHAR (128)	Идентификатор создателя таблицы, представления или псевдонима
TYPE	CHAR (1)	Тип: 'T' — таблица, 'V' — представление, 'A' — псевдоним, 'H' — иерархическая таблица, 'S' — материализованный запрос, 'U' — типизированная таблица, 'W' — типизированное представление
STATUS	CHAR (1)	Статус объекта (используется системой)
DROPRULE	CHAR (1)	'N' — правила нет, 'R' — правило удаления RESTRICT
BASE_TABSCHEMA	VARCHAR (128)	Схема, содержащая базовую таблицу, на которую ссылается псевдоним
BASE_TABNAME	VARCHAR (128)	Имя базовой таблицы, на которую ссылается псевдоним
ROWTYPESCHEMA	VARCHAR (128)	Имя схемы для типа строк данной таблицы
ROWTYPENAME	VARCHAR (18)	Имя типа строк данной таблицы
CREATE_TIME	TIMESTAMP	Время создания объекта
STATS_TIME	TIMESTAMP	Время, когда последний раз вычислялись статистические данные объекта
COLCOUNT	SMALLINT	Количество столбцов в таблице
TABLEID	SMALLINT	Внутренний идентификатор таблицы
TBSPACEID	SMALLINT	Идентификатор первичного табличного пространства, в котором хранится таблица
CARD	INTEGER	Количество записей в таблице
NPAGES	INTEGER	Число страниц дисковой памяти, содержащих данные таблицы
FPAGES	INTEGER	Общее число страниц дисковой памяти, занимаемых таблицей
OVERFLOW	INTEGER	Число переполнения записей таблицы
TBSPACE	VARCHAR (18)	Первичное табличное пространство таблицы
INDEX_TBSPACE	VARCHAR (18)	Табличное пространство, в котором хранятся индексы таблицы
LONG_TBSPACE	VARCHAR (18)	Табличное пространство, в котором хранятся большие двоичные объекты
PARENTS	SMALLINT	Число родительских таблиц для данной таблицы
CHILDREN	SMALLINT	Число дочерних таблиц для данной таблицы

Столбец	Тип данных	Информация
SELFREFS	SMALLINT	Число ссылок таблицы на саму себя
KEYCOLUMNS	SMALLINT	Число столбцов в первичном ключе таблицы
KEYINDEXID	SMALLINT	Внутренний идентификатор первичного ключа таблицы
KEYUNIQUE	SMALLINT	Число ограничений уникальности на значения столбцов таблицы
CHECKCOUNT	SMALLINT	Число ограничений на значения столбцов таблицы
DATA_CAPTURE	CHAR(1)	Признак реплицированной таблицы
CONST_CHECKED	CHAR(32)	Флаги проверки ограничений
PMAP_ID	SMALLINT	Внутренний идентификатор схемы физической сегментации таблицы
PARTITION_MODE	CHAR(1)	Режим сегментации таблиц базы данных
LOG_ATTRIBUTE	CHAR(1)	Признак того, что для таблицы по умолчанию установлен режим ведения журнала
PCTFREE	SMALLINT	Часть страниц (в процентах), которую следует зарезервировать для будущих данных
REMARKS	VARCHAR(254)	Пользовательские примечания к таблице

С помощью таких запросов, как приведенные ниже, можно получить информацию о таблицах в СУБД DB2. Используя аналогичные запросы с другими именами таблиц и столбцов, можно получить ту же самую информацию в СУБД других типов.

*Имена всех таблиц базы данных, а также имена владельцев этих таблиц.*

```
SELECT DEFINER, TABNAME
FROM SYSCAT.TABLES
WHERE TYPE = 'T';
```

*Имена всех таблиц, представлений и псевдонимов, имеющихся в базе данных.*

```
SELECT TABNAME
FROM SYSCAT.TABLES;
```

*Имена и даты создания моих таблиц.*

```
SELECT TABNAME, CREATE_TIME
FROM SYSCAT.TABLES
WHERE TYPE = 'T'
AND DEFINER = USER;
```

В СУБД Oracle аналогичные функции выполняют системные представления ALL\_TABLES, DBA\_TABLES и USER\_TABLES, описанные в табл. 16.3. Представление ALL\_TABLES содержит по одной строке для каждой таблицы, к которой текущий пользователь имеет по крайней мере одну привилегию доступа. Представление DBA\_TABLES содержит по строке для каждой таблицы базы данных. DBA-представления (т.е. те, имена которых начинаются на DBA\_) обычно доступны только тем пользователям, которые имеют в базе данных высший уровень привилегий, таким как администратор базы данных. Представление USER\_TABLES со-

держит по одной строке для каждой таблицы, принадлежащей текущему пользователю. Все три представления содержат одинаковые столбцы, за исключением столбца OWNER, отсутствующего в представлении USER\_TABLES.

**Таблица 16.3. Избранные столбцы представления каталога ALL\_TABLES, DBA\_TABLES и USER\_TABLES СУБД Oracle**

Столбец	Тип данных	Информация
OWNER	VARCHAR2 (30)	Владелец таблицы (отсутствует в USER_TABLES)
TABLE_NAME	VARCHAR2 (30)	Имя таблицы
TABLESPACE_NAME	VARCHAR2 (30)	Имя табличного пространства, содержащего таблицу; NULL для сегментированных, временных и индексно-организованных таблиц
CLUSTER_NAME	VARCHAR2 (30)	Имя кластера, к которому относится таблица (если таковой имеется)
IOT_NAME	VARCHAR2 (30)	Имя индексно-организованной таблицы (если таковая имеется), к которой относятся записи с переполнением или таблицы отображения
STATUS	VARCHAR2 (8)	Указывает, корректна таблица (VALID) или нет (UNUSABLE)
PCT_FREE	NUMBER	Минимальный процент свободного пространства в блоке; NULL для секционированных таблиц
PCT_USED	NUMBER	Минимальный процент используемого пространства в блоке; NULL для секционированных таблиц
INI_TRANS	NUMBER	Изначальное количество транзакций; NULL для секционированных таблиц
MAX_TRANS	NUMBER	Максимальное количество транзакций; NULL для секционированных таблиц
INITIAL_EXTENT	NUMBER	Размер начального экстенда в байтах; NULL для секционированных таблиц
NEXT_EXTENT	NUMBER	Размер вторичных экстендов в байтах; NULL для секционированных таблиц
MIN_EXTENTS	NUMBER	Минимально допустимое количество экстендов в сегменте; NULL для секционированных таблиц
MAX_EXTENTS	NUMBER	Максимально допустимое количество экстендов в сегменте; NULL для секционированных таблиц
NUM_ROWS	NUMBER	Количество строк в таблице (NULL при обновлении статистики)
BLOCKS	NUMBER	Количество используемых блоков данных в таблице (NULL при обновлении статистики)
EMPTY_BLOCKS	NUMBER	Количество блоков в таблице, которые никогда не использовались (NULL при обновлении статистики)
AVG_SPACE	NUMBER	Среднее количество свободного пространства в байтах в блоке данных, выделенном таблице
PARTITIONED	VARCHAR2 (3)	Указывает, секционирована таблица (YES) или нет (NO)
TEMPORARY	VARCHAR2 (1)	Указывает, временная это таблица (Y) или нет (N)



Вот типичные запросы к представлениям системного каталога Oracle.

*Имена и владельцы всех таблиц, к которым имеет доступ текущий пользователь.*

```
SELECT TABLE_NAME, OWNER  
FROM ALL_TABLES;
```

*Имена и владельцы всех таблиц базы данных.*

```
SELECT TABLE_NAME, OWNER  
FROM DBA_TABLES;
```

*Имена всех таблиц текущего пользователя.*

```
SELECT TABLE_NAME  
FROM USER_TABLES;
```

В СУБД SQL Server эквивалентом представления SYSCAT.TABLES из DB2 является представление каталога SYS.OBJECTS (табл. 16.4). Представление SYS.OBJECTS хранит информацию о таблицах, представлениях и других объектах SQL Server, таких как хранимые процедуры, правила и триггеры. Обратите также внимание на то, что для идентификации владельца таблицы в представлении SYS.OBJECTS используется внутренний идентификатор пользователя (principal\_id) вместо его имени.

**Таблица 16.4. Столбцы представления SYS.OBJECTS каталога SQL Server**

Имя столбца	Тип данных	Информация
Name	sysname	Имя объекта
object_id	int	Внутренний идентификатор объекта
schema_id	int	Идентификатор схемы, содержащей объект
principal_id	int	Идентификатор владельца индивидуального объекта (если он отличается от владельца схемы)
parent_object_id	int	Идентификатор объекта, которому принадлежит данный объект (0, если объект не является дочерним)
type	char(2)	Тип объекта: C — ограничение CHECK D — ограничение DEFAULT F — ограничение FOREIGN KEY P — хранимая процедура PK — ограничение PRIMARY KEY S — системная таблица TR — триггер U — пользовательская таблица (плюс много других значений)
create_date	datetime	Дата и время создания объекта
modify_date	datetime	Дата и время последнего изменения объекта
is_ms_shipped	bit	Был объект создан внутренним компонентом SQL Server или нет
is_published	bit	Опубликован объект или нет
is_schema_published	bit	Опубликована только схема объекта или нет

В Informix Universal Server системная таблица, содержащая информацию о таблицах базы данных, называется SYSTABLES. Как и каталог DB2, она содержит информацию только о таблицах, представлениях и псевдонимах; другие объекты базы данных описываются в других системных таблицах. Вот типичный запрос к системной таблице Informix.

*Имена, владельцы и даты создания всех таблиц базы данных.*

```
SELECT TABNAME, OWNER, CREATED
FROM SYSTABLES
WHERE TABTYPE = 'T';
```

Как показывают эти примеры, запросы на получение информации о таблицах имеют похожую структуру в различных СУБД. Тем не менее конкретные имена системных таблиц или представлений, а также их столбцов могут существенно отличаться.

## Информация о столбцах

Во всех основных реляционных СУБД имеется системная таблица, в которой отслеживается состояние столбцов базы данных. В этой таблице отводится одна строка для каждого столбца каждой таблицы или представления базы данных. Большинство СУБД ограничивает доступ пользователей к этой таблице, вместо этого создавая системное представление, описывающее столбцы только тех таблиц, которыми владеет пользователь или к которым он имеет доступ. В Oracle эта информация содержится в трех представлениях системного каталога — USER\_TAB\_COLUMNS, включающем одну строку для каждого столбца в каждой таблице, принадлежащей текущему пользователю; ALL\_TAB\_COLUMNS, включающем одну строку для каждого столбца каждой таблицы, для работы с которой текущему пользователю предоставлена хотя бы одна привилегия доступа; и DBA\_TAB\_COLUMNS, включающем одну строку для каждого столбца каждой таблицы базы данных.

Большая часть информации в описываемой системной таблице или представлении относится к определению столбца. Здесь указывается его имя, тип данных, длина, могут ли в нем присутствовать значения NULL и т.д. Кроме того, иногда в таблице приводится информация о том, как распределены значения данных в каждом столбце. Эта статистическая информация помогает СУБД выполнять запросы оптимальным образом.

С помощью запросов, аналогичных приведенному ниже, можно получать информацию о столбцах в СУБД Oracle.

*Имена и типы данных столбцов моей таблицы OFFICES.*

```
SELECT COLUMN_NAME, DATA_TYPE
FROM USER_TAB_COLUMNS
WHERE TABLE_NAME = 'OFFICES';
```

**Таблица 16.5. Избранные столбцы представления SYSCAT.COLUMNS (СУБД DB2)**

Имя столбца	Тип данных	Информация
TABSCHEMA	VARCHAR (128)	Схема, которой принадлежит таблица, содержащая столбец
TABNAME	VARCHAR (128)	Имя таблицы, содержащей столбец
COLNAME	VARCHAR (128)	Имя столбца
COLNO	SMALLINT	Позиция столбца в таблице (первый столбец — 0)
TYPESCHEMA	VARCHAR (128)	Схема, которой принадлежит домен столбца (по умолчанию SYSIBM)
TYPENAME	VARCHAR (18)	Название типа данных или домена столбца
LENGTH	INTEGER	Максимальная длина для столбцов, содержащих значения переменной длины
SCALE	SMALLINT	Масштаб чисел типа DECIMAL
DEFAULT	VARCHAR (254)	Значение по умолчанию
NULLS	CHAR (1)	'Y' — разрешены значения NULL; 'N' — не разрешены
CODEPAGE	SMALLINT	Кодовая страница для текстовых данных, представленных в расширенной кодировке
LOGGED	CHAR (1)	'Y' — разрешено ведение журнала доступа к столбцам, содержащим большие двоичные объекты; 'N' — ведение журнала не разрешено
COMPACT	CHAR (1)	'Y' — столбец, содержащий большие двоичные объекты, является сжатым; 'N' — столбец не сжат
COLCARD	BIGINT	Число различных значений в столбце
HIGH2KEY	VARCHAR (254)	Второе наибольшее значение данных в столбце
LOW2KEY	VARCHAR (254)	Второе наименьшее значение данных в столбце
AVGCOLLEN	INTEGER	Средняя длина для столбцов, содержащих значения переменной длины
KEYSEQ	SMALLINT	Позиция столбца в первичном ключе (или 0)
PARTKEYSEQ	SMALLINT	Позиция столбца в ключе сегментирования (или 0)
NQUANTILES	SMALLINT	Число квантилей в статистике столбца
NMOSTFREQ	SMALLINT	Число часто встречающихся значений в статистике столбца
REMARKS	VARCHAR (254)	Пользовательские примечания к столбцу

Подобно информации в системном каталоге, информация о столбцах представлена по-разному в различных СУБД. В табл. 16.5 описано содержимое системной таблицы SYSCAT.COLUMNS, в которой помещаются сведения о столбцах базы данных в DB2. Вот несколько запросов, которые можно выполнить в этой СУБД.

*Найти в базе данных все столбцы с типом данных DATE.*

```
SELECT TABSCHEMA, TABNAME, COLNAME
FROM SYSCAT.COLUMNS
WHERE TYPESCHEMA = 'SYSIBM'
AND TYPENAME = 'DATE';
```

Указать имя владельца, имя столбца, тип данных и длину для всех содержащих более десяти символов текстовых столбцов представлений.

```
SELECT DEFINER, COLS.TABNAME, COLNAME, TYPENAME, LENGTH
FROM SYSCAT.COLUMNS COLS, SYSCAT.TABLES TBLS
WHERE TBLS.TABSCHEMA = COLS.TABSCHEMA
AND TBLS.TABNAME = COLS.TABNAME
AND (TYPENAME = 'VARCHAR' OR TYPENAME = 'CHARACTER')
AND LENGTH > 10
AND TYPE = 'V';
```

Определения столбцов в системных каталогах СУБД различных типов значительно отличаются друг от друга. Для сравнения в табл. 16.6 приведено определение таблицы SYSCOLUMNS из Informix Universal Server. Некоторые различия между таблицами, содержащими описания столбцов, носят лишь стилистический характер.

- Имена столбцов в указанных таблицах различны, даже когда они содержат аналогичные данные.
- Для идентификации таблицы, содержащей данный столбец, в DB2 используется комбинация имен схемы и таблицы; в Informix для этой цели применяется внутренний идентификатор, который является внешним ключом для таблицы SYSTABLES.
- В системном каталоге DB2 типы данных задаются в текстовом виде (например, CHARACTER); в системном каталоге Informix используются целочисленные коды типов данных.

**Таблица 16.6. Таблица SYSCOLUMNS (СУБД Informix)**

Имя столбца	Тип данных	Информация
COLNAME	VARCHAR (128)	Имя столбца
TABID	INTEGER	Внутренний идентификатор таблицы, в которой содержится столбец
COLNO	SMALLINT	Позиция столбца в таблице
COLTYPE	SMALLINT	Код типа данных столбца, а также разрешены ли в нем значения NULL
COLLENGTH	SMALLINT	Длина столбца в байтах
COLMIN	INTEGER	Минимальная длина столбца в байтах
COLMAX	INTEGER	Максимальная длина столбца в байтах
EXTENDED_ID	INTEGER	Внутренний идентификатор расширенного типа данных
SECLABLID	INTEGER	Идентификатор метки безопасности столбца

Другие различия отражают разные возможности этих двух СУБД.

- DB2 позволяет делать примечания (длиной до 254 символов) к каждому столбцу; в Informix такой возможности нет.

- В Informix отслеживается минимальный и максимальный размер данных для столбцов, хранящих значения переменной длины; в DB2 такая информация непосредственно недоступна.

## Информация о представлениях

Определения представлений, имеющихся в базе данных, хранятся в системном каталоге. Каталог СУБД DB2 содержит две системные таблицы, в которых отслеживается состояние представлений. Таблица SYSCAT.VIEWS (табл. 16.7) содержит SQL-определения всех представлений в текстовом виде. Старые версии DB2 поддерживали текст SQL длиной до 3600 символов, и в них определения, превышающие этот размер, хранились в нескольких последовательно пронумерованных строках (1, 2, 3 и т.д.). В новых версиях DB2 используется тип данных CLOB, способный принимать определения размером до 64 Кбайт, так что для каждого представления достаточно одной строки таблицы SYSCAT.VIEWS.

**Таблица 16.7. Присваивание SYSCAT.VIEWS (СУБД DB2)**

Имя столбца	Тип данных	Информация
VIEWSCHEMA	VARCHAR (128)	Схема, содержащая представление
VIEWNAME	VARCHAR (128)	Имя представления
DEFINER	VARCHAR (128)	Идентификатор создателя представления
SEQNO	SMALLINT	Последовательный номер данной строки SQL-текста (в DB2 UDB всегда 1)
VIEWCHECK	CHAR (1)	Тип проверки представления: 'N' — проверки нет 'L' — локальная проверка 'C' — каскадная проверка
READONLY	CHAR (1)	'Y', если представление доступно только для чтения; 'N' — в противном случае
VALID	CHAR (1)	'Y', если определение представления корректно; 'N' — в противном случае
QUALIFIER	VARCHAR (128)	Имя схемы по умолчанию в момент определения объекта
FUNC_PATH	VARCHAR (254)	Путь для разрешения вызовов функций в представлении
TEXT	CLOB (64K)	SQL-текст определения представления ("SELECT . . ."); в старых версиях DB2 тип данных VARCHAR (3600)

С помощью этой таблицы можно посмотреть определения всех представлений базы данных. Как и в большинстве основных коммерческих СУБД, в DB2 информация о представлениях тесно связана с информацией о таблицах в каталоге DB2. Это означает, что нужные сведения часто можно получить несколькими способами. Вот, например, непосредственный запрос к системной таблице DB2 VIEWS, возвращающий имена всех представлений базы данных вместе с именами их создателей.

*Список всех представлений, определенных в базе данных.*

```
SELECT DISTINCT VIEWSHEMA, VIEWNAME, DEFINER  
FROM SYSCAT.VIEWS;
```

Обратите внимание на применение предиката DISTINCT в старых версиях DB2 для удаления из списка тех представлений, определения которых слишком длинны для размещения в одной строке. Пожалуй, более простой способ получить ту же информацию — обратиться непосредственно к системной таблице TABLES, отбирая только те строки, которые соответствуют представлениям, согласно столбцу TYPE.

*Список всех представлений, определенных в базе данных.*

```
SELECT TABSCHEMA, TABNAME, DEFINER  
FROM SYSCAT.TABLES  
WHERE TYPE = 'V';
```

В большинстве ведущих СУБД представления описываются в структуре системного каталога схожим образом. В Informix Universal Server, например, имеется системная таблица SYSVIEWS, содержащая описания представлений. В каждой ее строке хранится 64-символьный фрагмент инструкции SELECT, формирующей представление. Если представление занимает несколько фрагментов, каждому из них присваивается порядковый номер, как и в DB2. В таблице SYSVIEWS содержится лишь один дополнительный столбец, в котором находятся идентификаторы таблиц, связывающие таблицу SYSVIEWS с соответствующими записями таблицы SYSTABLES. Таким образом дублируется меньше информации, но вам придется выполнять явное соединение обеих системных таблиц, чтобы получить информацию о представлении.

В Oracle SQL-текст определения представления также доступен через системное представление. Как и в случае таблиц и столбцов, имеется три системных представления — USER\_VIEWS, содержащее информацию обо всех представлениях, принадлежащих текущему пользователю; ALL\_VIEWS, содержащее информацию обо всех представлениях, к которым текущий пользователь имеет доступ; и DBA\_VIEWS, содержащее информацию обо всех представлениях базы данных. В Oracle текст определения представления хранится в столбце типа LONG (специфичный для Oracle тип данных для хранения больших текстов) и может содержать много тысяч символов. Имеется также столбец TEXT\_LENGTH, в котором записана длина этого определения. Вот как в СУБД Oracle можно получить информацию о представлениях.

*Определения всех представлений, принадлежащих текущему пользователю.*

```
SELECT VIEW_NAME, TEXT_LENGTH, TEXT  
FROM USER_VIEWS;
```

Заметим, что в большинстве интерактивных продуктов SQL (включая Oracle) при выводе на экран длинный текст обрезается. В базе данных же хранится полный текст.

## Примечания

В СУБД DB2 вы можете создавать *примечания* (длиной до 254 символов) для каждой таблицы, каждого представления или столбца базы данных. Примечания позволяют сохранить в системном каталоге краткое описание таблицы или других данных. Примечания хранятся в системных таблицах SYSCAT.TABLES и SYSCAT.COLUMNS. В отличие от других элементов определения таблицы или столбца, примечания не задаются в инструкции CREATE TABLE. Вместо этого следует использовать инструкцию COMMENT (рис. 16.2). Ниже приведено несколько примеров использования инструкции COMMENT.

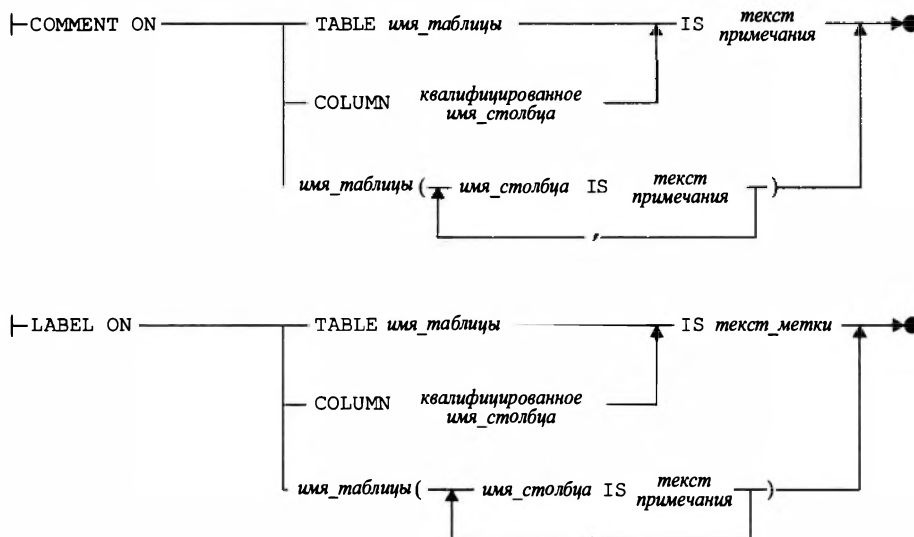


Рис. 16.2. Синтаксическая диаграмма инструкции COMMENT в DB2

Создать примечание для таблицы OFFICES.

```
COMMENT ON TABLE OFFICES
  IS 'Эта таблица содержит данные о наших офисах';
```

Создать примечания для столбцов TARGET и SALES таблицы OFFICES.

```
COMMENT ON OFFICES
(TARGET IS 'Это плановый годовой объем продаж офиса',
 SALES IS 'Это текущий объем продаж офиса');
```

Поскольку эта возможность берет свое начало из самых ранних продуктов SQL фирмы IBM, Oracle также поддерживает инструкцию COMMENT ON для назначения комментариев таблицам и столбцам. Однако эта информация хранится отдельно от прочей информации о таблицах и столбцах. Она доступна посредством системных представлений Oracle USER\_TAB\_COMMENTS и USER\_COL\_COMMENTS. В DB2 разрешается также создавать примечания для ограничений, хранимых процедур, схем, табличных пространств, триггеров и других объектов базы данных. Эта возможность не поддерживается ни в стандарте SQL, ни в большинстве других СУБД.

## Информация об отношениях между таблицами

В середине 90-х годов, наряду с введением понятия ссылочной целостности, в ведущих корпоративных СУБД системные каталоги были расширены информацией о первичных ключах, внешних ключах и создаваемых ими отношениях “предок-потомок”. В DB2, которая была одной из первых СУБД, поддерживающих ссылочную целостность, эта информация находится в системной таблице SYSCAT.REFERENCES (табл. 16.8). Каждое отношение “предок-потомок” между двумя таблицами базы данных представлено одной строкой. В этой строке содержатся имена родительской и дочерней таблиц, имя отношения, а также правила удаления и обновления для этого отношения. Чтобы получить информацию об отношениях в базе данных, вам следует выполнить запрос к этой таблице.

*Список всех отношений “предок-потомок” между моими таблицами, включая имя отношения, имя таблицы-предка, имя таблицы-потомка и правила удаления и обновления для каждого отношения.*

```
SELECT CONSTNAME, REFTABNAME, TABNAME,  
       DELETERULE, UPDATERULE  
FROM SYSCAT.REFERENCES  
WHERE DEFINER = USER;
```

*Список всех таблиц, связанных с таблицей SALESREPS (как предков, так и потомков).*

```
SELECT REFTABNAME  
FROM SYSCAT.REFERENCES  
WHERE TABNAME = 'SALESREPS'  
UNION  
SELECT TABNAME  
FROM SYSCAT.REFERENCES  
WHERE REFTABNAME = 'SALESREPS';
```

**Таблица 16.8. Представление SYSCAT.REFERENCES (СУБД DB2)**

Имя столбца	Тип данных	Информация
CONSTNAME	VARCHAR (128)	Имя отношения, описываемого данной строкой
TABSCHEMA	VARCHAR (128)	Схема, содержащая данное именованное отношение
TABNAME	VARCHAR (128)	Имя таблицы, к которой применяется ограничение
OWNER	VARCHAR (128)	Создатель таблицы, к которой применяется ограничение
REFKEYNAME	VARCHAR (128)	Имя родительского ключа
REFTABSCHEMA	VARCHAR (128)	Схема, содержащая родительскую таблицу
REFTABNAME	VARCHAR (128)	Имя родительской таблицы
COLCOUNT	SMALLINT	Число столбцов во внешнем ключе
DELETERULE	CHAR (1)	Правило удаления для ограничения внешнего ключа ('A' — отсутствие правила, 'C' — CASCADE, 'R' — RESTRICT)
UPDATERULE	CHAR (1)	Правило обновления для ограничения внешнего ключа ('A' — отсутствие правила, 'R' — RESTRICT)



Имя столбца	Тип данных	Информация
CREATE_TIME	TIMESTAMP	Время создания ограничения
FK_COLNAMES	VARCHAR (640)	Названия столбцов внешнего ключа
PK_COLNAMES	VARCHAR (640)	Названия столбцов первичного ключа
DEFINER	VARCHAR (128)	Идентификатор авторизации, под которым было создано данное ограничение

Имена столбцов внешнего ключа и соответствующих им столбцов первичного ключа перечислены в текстовом виде в столбцах FK\_COLNAMES и PK\_COLNAMES таблицы SYSCAT.REFERENCES. Более полезная информация содержится в другой системной таблице — SYSCAT.KEYCOLUSE (табл. 16.9). Для каждого столбца каждого внешнего ключа, первичного ключа и ограничения уникальности в этой системной таблице отводится одна строка. Специальная последовательность чисел определяет порядок следования столбцов в составном внешнем ключе. С помощью запроса к этой таблице, подобного приведенному ниже, можно узнать имена столбцов, связывающих некоторую таблицу с ее предком.

*Список столбцов, связывающих таблицу ORDERS с таблицей PRODUCTS в отношении ISFOR.*

```
SELECT COLNAME, COLSEQ
FROM SYSCAT.KEYCOLUSE
WHERE CONSTNAME = 'ISFOR'
ORDER BY COLSEQ;
```

**Таблица 16.9. Представление SYSCAT.KEYCOLUSE (СУБД DB2)**

Имя столбца	Тип данных	Информация
CONSTNAME	VARCHAR (128)	Имя описываемого строкой ограничения (ограничения уникальности, первичного ключа или внешнего ключа)
TABSCHEMA	VARCHAR (128)	Схема, содержащая данное ограничение
TABNAME	VARCHAR (128)	Имя таблицы, к которой применяется ограничение
COLNAME	VARCHAR (128)	Имя столбца, участвующего в ограничении
COLSEQ	SMALLINT	Позиция, занимаемая столбцом в ограничении (первый столбец = 1)

Информация о первичных ключах и отношениях “предок-потомок”, в которых они участвуют, содержится также в системных таблицах SYSCAT.TABLES и SYSCAT.COLUMNS, описанных ранее в табл. 16.2 и 16.5. Если таблица имеет первичный ключ, то ее строка в таблице SYSCAT.TABLES имеет в столбце KEYCOLUMNS значение, отличное от нуля и равное числу столбцов первичного ключа (1 — для простого ключа, 2 и более — для составного ключа). В таблице SYSCAT.COLUMNS строки столбцов, входящих в состав первичных ключей, имеют в столбце KEYSEQ значение, не равное нулю. Оно указывает позицию (1, 2 и т.д.), занимаемую столбцом в первичном ключе.

Чтобы найти первичный ключ какой-либо таблицы, вы можете сделать следующий запрос к таблице SYSCAT.COLUMNS.

*Список столбцов, образующих первичный ключ таблицы PRODUCTS.*

```
SELECT COLNAME, KEYSEQ, TYPENAME, REMARKS
FROM SYSCAT.COLUMNS
WHERE TABNAME = 'PRODUCTS'
AND KEYSEQ > 0
ORDER BY KEYSEQ;
```

В системных каталогах других основных СУБД поддержка первичных и внешних ключей осуществляется схожим образом. Например, в Oracle имеются системные представления ALL\_CONSTRAINTS и USER\_CONSTRAINTS, предоставляющие ту же информацию, что и системная таблица DB2 SYSCAT.REFERENCES. Информация о столбцах внешних и первичных ключей содержится в системных представлениях ALL\_CONS\_COLUMNS и USER\_CONS\_COLUMNS, которые аналогичны таблице SYSCAT.KEYCOLUSE DB2. В Microsoft SQL Server структура системного каталога подобна описанной выше, а информация о внешних ключах распределена между представлениями SYS.FOREIGN\_KEYS и SYS.FOREIGN\_KEY\_COLUMNS.

В Informix Universal Server применяется подход, близкий используемому в DB2, но с теми же отличиями, которые уже были описаны при рассмотрении представления в системном каталоге информации о столбцах. Для каждого ограничения, определенного в базе данных, создается строка в системной таблице SYSCONSTRAINTS, где записано имя ограничения и его тип (условие на значение, первичный ключ, внешний ключ и т.д.). В этой таблице ограничению назначается также внутренний идентификатор, используемый при ссылке на данное ограничение в других таблицах системного каталога. Таблица, к которой применяется ограничение, тоже идентифицируется внутренним номером (он служит в качестве внешнего ключа к таблице SYSTABLES).

Более подробная информация о ссылочной целостности (внешних ключах) содержится в таблице SYSREFERENCES. В ней ограничение, первичный ключ и родительская таблица также определяются внутренними идентификаторами, создающими связи с таблицами SYSCONSTRAINTS и SYSTABLES. В таблице SYSREFERENCES описаны правила удаления и обновления для отношения, создаваемого внешним ключом, и другая подобная информация.

## Информация о пользователях

В общем случае системный каталог содержит таблицу, в которой перечислены все пользователи, имеющие санкционированный доступ к базе данных. СУБД может использовать эту системную таблицу для проверки имени и пароля пользователя, когда он первый раз устанавливает соединение с базой данных. В таблице может также храниться и другая информация о пользователях.

В SQL Server информация о пользователях хранится в системном представлении SYS.DATABASE\_PRINCIPALS (табл. 16.10). Каждая строка этой таблицы описывает одного пользователя или группу пользователей, входящих в схему защиты ба-

зы данных. В Informix применяется схожий подход; соответствующая системная таблица тоже называется SYSUSERS. В Oracle эта таблица называется DBA\_USERS. Вот два эквивалентных запроса к SQL Server и Oracle, которые выводят список зарегистрированных пользователей.

*Вывести все идентификаторы пользователей, зарегистрированных в SQL Server.*

```
SELECT NAME
FROM SYS.DATABASE_PRINCIPALS;
```

*Вывести все идентификаторы пользователей, зарегистрированных в Oracle.*

```
SELECT USERNAME
FROM DBA_USERS;
```

**Таблица 16.10.** Столбцы представления каталога SYS.DATABASE\_PRINCIPALS (СУБД SQL Server)

Имя столбца	Тип данных	Информация
name	sysname	Уникальное в пределах базы данных имя пользователя
principal_id	int	Уникальный в пределах базы данных идентификатор пользователя
type	char(1)	Тип пользователя: S — пользователь SQL U — пользователь Windows G — группа Windows A — роль приложения R — роль базы данных C — пользователь, отображенный на сертификат K — пользователь, отображенный на асимметричный ключ
type_desc	nvarchar(60)	Описание типа пользователя
default_schema_name	sysname	Имя, используемое в случае, когда имя SQL не определяет схему
create_date	datetime	Дата и время создания пользователя
modify_date	datetime	Дата и время последней модификации пользователя
owning_principal_id	int	Идентификатор владельца данного пользователя
sid	varbinary(85)	Идентификатор безопасности (Security identifier, SID), если пользователь определен как внешний по отношению к базе данных (типы S, U или G)
is_fixed_role	bit	Если значение равно 1, то строка представляет запись для одной из фиксированных ролей, таких как db_owner

В DB2 таблица SYSCAT.DBAUTH, содержащая имена пользователей, хранит также информацию о ролях и привилегиях пользователей в базе данных (т.е. являются ли они администраторами базы данных, могут ли создавать таблицы, могут ли

создавать программные модули для доступа к базе данных и т.д.). Вот запрос к СУБД DB2, эквивалентный показанным выше.

*Вывести все идентификаторы пользователей, зарегистрированных в DB2.*

```
SELECT DISTINCT GRANTEE  
FROM SYSCAT.DBAUTH  
WHERE GRANTEEType = 'U';
```

## Информация о привилегиях

Помимо информации о структуре базы данных, системный каталог в общем случае хранит информацию, которая необходима СУБД для обеспечения безопасности базы данных. Как уже говорилось в главе 15, “SQL и безопасность”, в разных СУБД применяются различные варианты базовой схемы выдачи привилегий SQL. Это находит свое отражение в структуре системных каталогов различных СУБД.

DB2 имеет одну из наиболее сложных схем пользовательских привилегий, которая охватывает даже отдельные столбцы таблиц. В табл. 16.11 приведены названия таблиц системного каталога DB2, которые хранят информацию о привилегиях, и кратко описана роль каждой из них.

**Таблица 16.11. Представления системного каталога DB2, реализующие привилегии**

Системная таблица	Роль
TABAUTH	Реализует привилегии на уровне таблиц, показывая, каким пользователям к каким таблицам и для каких операций (SELECT, INSERT, DELETE, UPDATE, ALTER и INDEX) разрешен доступ
COLAUTH	Реализует привилегии на уровне столбцов, показывая, какие пользователи имеют разрешение на обновление и создание внешних ссылок, а также на какие именно столбцы каких именно таблиц
DBAUTH	Определяет, какие пользователи имеют право устанавливать соединение с базой данных, создавать таблицы и выполнять различные функции администрирования базы данных
SCHEMAAUTH	Реализует привилегии на уровне схемы, показывая, какие пользователи имеют разрешения создавать, удалять или модифицировать объекты (таблицы, представления, домены и т.д.), входящие в схему
INDEXAUTH	Реализует привилегии на уровне индексов, показывая, какие пользователи имеют привилегии управления различными индексами
PACKAGEAUTH	Реализует привилегии программного доступа, показывая, какие пользователи могут создавать различные утилиты доступа к базам данных (“пакеты”), выполнять их и управлять ими

Схема авторизации пользователей, используемая в SQL Server, более фундаментальная и четкая, чем в DB2. Базы данных, таблицы, хранимые процедуры, триггеры и другие элементы рассматриваются в ней однотипно, как объекты, по отношению к которым назначаются привилегии. Четкая структура этой схемы отражена в представлении каталога SYS.DATABASE\_PERMISSIONS (табл. 16.12), которое реализует всю схему привилегий в SQL Server. Каждая строка этого представления соответствует одной инструкции GRANT или REVOKE, выполненной в базе данных.

**Таблица 16.12. Столбцы представления SYS.DATABASE\_PERMISSIONS (СУБД SQL Server)**

Имя столбца	Тип данных	Информация
class	tinyint	Класс, для которого имеется разрешение 0 — база данных 1 — объект или столбец 3 — схема 4 — пользователь базы данных 5 — пакет 6 — тип 10 — набор XML схем 15 — тип сообщения 16 — контракт службы 17 — служба 18 — подключение к удаленному сервису 19 — маршрут 23 — полнотекстовый каталог 24 — симметричный ключ 25 — сертификат 26 — асимметричный ключ
class_desc	nvarchar(60)	Описание класса, для которого имеется разрешение
major_id	int	Идентификатор сущности, для которой имеется разрешение
minor_id	int	Вторичный идентификатор сущности, для которой имеется разрешение
grantee_principal_id	int	Идентификатор пользователя базы данных, которому предоставлено разрешение
grantor_principal_id	int	Идентификатор пользователя базы данных, предоставившего разрешение
type	char(4)	Тип разрешения базы данных
permission_name	sysname	Имя разрешения
state	char(1)	Состояние разрешения
state_desc	nvarchar(60)	Описание состояния разрешения

## Информационная схема SQL

В стандарте SQL форма системного каталога, которую должны поддерживать реляционные СУБД, явно не определена. В то время, когда принимался стандарт SQL2, уже существовал широкий разброс характеристик коммерческих СУБД различных типов и имели место огромные различия в их системных каталогах, так что невозможно было достичь согласия по вопросу стандартной спецификации системного каталога. Вместо этого авторы стандарта дали определение идеализированного системного каталога, который поставщики СУБД могли бы применять при разработке с нуля СУБД, соответствующих стандарту SQL. Поскольку СУБД MySQL разрабатывалась после того, как SQL Information Schema вошла в стандарт SQL, она соответствует стандарту в этой части. Microsoft добавила в SQL Server 2008 ряд представлений, совместимых с SQL Information Schema. Многие иные производители намерены поступить так же.

Таблицы этого идеализированного системного каталога (который в стандарте называется *схема определений*) приведены в табл. 16.13.

**Таблица 16.13. Избранные столбцы стандартной схемы определений**

<b>Системная таблица</b>	<b>Содержимое</b>
ASSERTIONS	Одна строка для каждого утверждения
AUTHORIZATIONS	Одна строка для каждого имени роли и одна строка для каждого идентификатора авторизации
CHARACTER_SETS	Одна строка для каждого дескриптора набора символов
CHECK_COLUMN_USAGE	Одна строка для каждого столбца, упомянутого в ограничении на значение, ограничении домена или утверждении
CHECK_CONSTRAINTS	Одна строка для каждого ограничения домена, ограничения на значения таблицы или каждого утверждения
CHECK_TABLE_USAGE	Одна строка для каждой таблицы, упомянутой в условиях отбора, ограничения на значение, ограничения домена или утверждения
COLLATIONS	Одна строка для каждого дескриптора порядка сортировки
COLUMN_PRIVILEGES	Одна строка для каждого дескриптора привилегии столбца
COLUMNS	Одна строка для каждого столбца в каждом определении таблицы или представления
DATA_TYPE_DESCRIPTOR	Одна строка для каждого домена или столбца, определенного с типом данных
DOMAIN_CONSTRAINTS	Одна строка для каждого ограничения домена
DOMAINS	Одна строка для каждого домена
KEY_COLUMN_USAGE	Одна или несколько строк для каждой строки в таблице TABLE_CONSTRAINTS, которая участвует в ограничениях на значение или ограничениях первичного или внешнего ключа
REFERENTIAL_CONSTRAINTS	Одна строка для каждой строки таблицы TABLE_CONSTRAINTS, которая участвует в ограничениях внешнего ключа
SCHEMATA	Одна строка для каждой схемы
TABLE_CONSTRAINTS	Одна строка для каждого ограничения таблицы, указанного в определении таблицы
TABLE_PRIVILEGES	Одна строка для каждой привилегии таблицы
TABLES	Одна строка для каждой таблицы или представления
TRIGGER_COLUMN_USAGE	Одна строка для каждого столбца, к которому обращается триггер
TRIGGER_TABLE_USAGE	Одна строка для каждой таблицы, к которой обращается триггер
TRIGGERS	Одна строка для каждого триггера
USAGE_PRIVILEGES	Одна строка для каждого дескриптора привилегий использования
USER_DEFINED_TYPES	Одна строка для каждого пользовательского типа данных
VIEW_COLUMN_USAGE	Одна строка для каждого столбца, к которому обращается представление
VIEW_TABLE_USAGE	Одна строка для каждой таблицы в определении каждого представления (если представление определено как запрос к нескольким таблицам, для каждой таблицы будет иметься отдельная строка)
VIEWS	Одна строка для каждой таблицы или представления

Стандарт SQL не требует, чтобы СУБД реально поддерживала описанные таблицы системного каталога, как и каталог вообще. Вместо этого в стандарте SQL определен ряд представлений, основанных на этих системных таблицах и описы-

вающих те объекты базы данных, которые доступны для текущего пользователя. (Эти представления каталога называются в стандарте *информационной схемой*.) Для того чтобы СУБД соответствовала стандарту SQL на уровне Intermediate или Full, она должна поддерживать эти представления. Такой подход дает пользователю стандартный способ получения информации о доступных ему объектах базы данных с помощью стандартных запросов к представлениям системного каталога. Следует отметить, что на начальном уровне (Entry Level) совместимости со стандартом SQL поддержка системных представлений не требуется.

На практике основные коммерческие реляционные СУБД постепенно переходят к поддержке информационной схемы SQL, обычно путем создания соответствующих представлений на основе таблиц своих собственных системных каталогов. Структура системных каталогов большинства СУБД достаточно близка к требуемой в стандарте; по крайней мере, 90 процентов требований, связанных с совместимостью со стандартом SQL, можно реализовать относительно просто. Реализация остальных 10 процентов существенно труднее, учитывая значительные различия между СУБД и то, что даже обобщенные системные представления затрагивают внутренние особенности реализации самой СУБД.

В результате полная поддержка представлений каталога SQL обычно реализуется в крупных новых версиях СУБД, вместе с соответствующими изменениями в ядре программного обеспечения СУБД. Представления каталога, требуемые стандартом SQL, приведены в табл. 16.14, вместе с кратким описанием информации, содержащейся в каждом представлении. Вот несколько примеров запросов, которые могут использоваться для получения информации о структуре базы данных из представлений системного каталога.

*Имена всех таблиц и представлений, владельцем которых является текущий пользователь.*

```
SELECT TABLE_NAME  
FROM TABLES;
```

*Имена, позиции и типы данных всех столбцов во всех представлениях.*

```
SELECT TABLE_NAME, COLUMN_NAME, ORDINAL_POSITION, DATA_TYPE  
FROM COLUMNS  
WHERE (COLUMNS.TABLE_NAME IN  
      (SELECT TABLE_NAME FROM VIEWS));
```

*Количество столбцов в таблице OFFICES.*

```
SELECT COUNT(*)  
FROM COLUMNS  
WHERE (TABLE_NAME = 'OFFICES');
```

Заметим, что в случае MySQL имена представлений должны быть квалифицированы именем схемы INFORMATION\_SCHEMA (например, INFORMATION\_SCHEMA.TABLES), если только вы уже не находитесь в этой базе данных.

```
SELECT TABLE_NAME  
FROM INFORMATION_SCHEMA.TABLES;
```

**Таблица 16.14. Представления системного каталога в соответствии со стандартом SQL**

Представление	Содержимое
ADMINISTRATIVE_ROLE_AUTHORIZATIONS	Одна строка для каждой авторизации роли, включающей WITH ADMIN OPTION
APPLICABLE_ROLES	Одна строка для каждой роли, применимой к текущему пользователю
ASSERTIONS	Одна строка для каждого утверждения, которым владеет текущий пользователь; содержит имя утверждения и его характеристики
ATTRIBUTES	Одна строка для каждого пользовательского типа данных, определенного в каталоге
CHARACTER_SETS	Одна строка для каждого определения набора символов, доступного текущему пользователю
CHECK_CONSTRAINT_ROUTINE_USAGE	Одна строка для каждой выполнимой подпрограммы, владельцем которой является текущий пользователь и от которой зависит ограничение домена, таблицы или утверждение
CHECK_CONSTRAINTS	Одна строка для каждого ограничения на значение столбца таблицы, которой владеет текущий пользователь
COLLATIONS	Одна строка для каждого определения порядка сортировки, доступного текущему пользователю
COLLATION_CHARACTER_SET_APPLICABILITY	Одна строка для каждого набора символов, к которому применима сортировка
COLUMN_COLUMN_USAGE	Одна строка для каждого генерируемого столбца, который зависит от базового столбца
COLUMN_DOMAIN_USAGE	Одна строка для каждого столбца, определенного как зависимый от домена
COLUMN_PRIVILEGES	Одна строка для каждой привилегии для работы со столбцом, предоставленной текущему пользователю или предоставленной им другому пользователю; содержит имя таблицы и столбца, тип привилегии, указание на то, кто предоставил привилегию, кому она предоставлена и имеет ли текущий пользователь право передачи этой привилегии
COLUMN_UDT_USAGE	Одна строка для каждого столбца, который зависит от пользовательского типа данных
COLUMNS	Одна строка для каждого столбца, доступного текущему пользователю; содержит имя столбца, имя таблицы или представлений, включающих в себя данный столбец, тип его данных и другую информацию
CONSTRAINT_COLUMN_USAGE	Одна строка для каждого столбца, на который имеется ссылка в условии на значение, условии уникальности, утверждении или определении внешнего ключа, принадлежащих текущему пользователю
CONSTRAINT_TABLE_USAGE	Одна строка для каждой таблицы, на которую имеется ссылка в условии на значение, условии уникальности, утверждении или определении внешнего ключа, принадлежащих текущему пользователю
DATA_TYPE_PRIVILEGES	Одна строка для каждого объекта схемы, который включает дескриптор типа данных, доступного данному пользователю или роли
DIRECT_SUPERTABLES	Одна строка для каждой непосредственной надтаблицы, связанной с таблицей, определенной в данном каталоге, и владельцем которой является данный пользователь или роль
DIRECT_SUPERTYPES	Одна строка для каждого непосредственного определенного в данном каталоге надтипа, владельцем которого является данный пользователь или роль



Представление	Содержимое
DOMAIN_CONSTRAINTS	Одна строка для каждого ограничения домена; содержит имя ограничения и его характеристики
DOMAINS	Одна строка для каждого домена, доступного текущему пользователю; содержит имя домена, базовый тип данных и его характеристики
ELEMENT_TYPES	Одна строка для каждого определенного в данном каталоге типа элемента, владельцем которого является данный пользователь или роль
ENABLED_ROLES	Одна строка для каждой роли, действующей в текущей сессии SQL
FIELDS	Одна строка для каждого определенного в данном каталоге типа поля, владельцем которого является данный пользователь или роль
INFORMATION_SCHEMA_CATALOG_NAME	Одна строка с именем базы данных для каждого пользователя ("каталога", по терминологии стандарта SQL), описываемого данной информационной схемой
KEY_COLUMN_USAGE	Одна строка для каждого столбца, на который наложено ограничение первичного или внешнего ключа либо ограничение уникальности и который входит в таблицу, принадлежащую текущему пользователю; строка содержит имя таблицы, имя столбца и позицию столбца в ключе
METHOD_SPECIFICATION_PARAMETERS	Одна строка для каждого SQL-параметра спецификации метода, определенной в представлении METHOD_SPECIFICATIONS
METHOD_SPECIFICATIONS	Одна строка для каждого SQL-метода каталога, доступного данному пользователю или роли
PARAMETERS	Одна строка для каждого SQL-параметра определенной в данном каталоге подпрограммы, который доступен данному пользователю или роли
REFERENCED_TYPES	Одна строка для каждого определенного в данном каталоге ссылочного типа, который доступен данному пользователю или роли
REFERENTIAL_CONSTRAINTS	Одна строка для каждого ссылочного ограничения (определения внешнего ключа) на таблицу, которой владеет текущий пользователь; содержит имена ограничений и имена родительских и дочерних таблиц
ROLE_COLUMN_GRANTS	Одна строка для каждой привилегии для работы с определенным в данном каталоге столбцом, который доступен (или права на который переданы) активным в настоящий момент ролям
ROLE_ROUTINE_GRANTS	Одна строка для каждой привилегии для работы с вызываемой SQL-подпрограммой, которая определена в данном каталоге и доступна (или права на нее переданы) активным в настоящий момент ролям
ROLE_TABLE_GRANTS	Одна строка для каждой привилегии для работы с определенной в данном каталоге таблицей, которая доступна (или права на которую переданы) активным в настоящий момент ролям
ROLE_TABLE_METHOD_GRANTS	Одна строка для каждой привилегии для работы с определенным над таблицами структурированных типов методом, который доступен (или права на который переданы) активным в настоящий момент ролям
ROLE_USAGE_GRANTS	Одна строка для каждой определенной в данном каталоге привилегии USAGE, которая доступна (или права на которую переданы) активным в настоящий момент ролям
ROLE_UDT_GRANTS	Одна строка для каждой привилегии для работы с определенным в данном каталоге пользовательским типом данных, который доступен (или права на который переданы) активным в настоящий момент ролям

Представление	Содержимое
ROUTINE_COLUMN_USAGE	Одна строка для каждого столбца, владельцем которого является текущий пользователь или роль и от которого зависят подпрограммы SQL, определенные в данном каталоге
ROUTINE_PRIVILEGES	Одна строка для каждой привилегии для работы с вызываемыми SQL-подпрограммами, которые определены в данном каталоге и доступны (или права на них переданы) активным в настоящий момент ролям
ROUTINE_ROUTINE_USAGE	Одна строка для каждой вызываемой SQL-подпрограммы, владельцем которой является данный пользователь или роль, от которой зависит SQL-подпрограмма, определенная в данном каталоге
ROUTINE_SEQUENCE_USAGE	Одна строка для каждого генератора внешней последовательности, владельцем которого является данный пользователь или роль и от которой зависит SQL-подпрограмма, определенная в данном каталоге
ROUTINE_TABLE_USAGE	Одна строка для каждой таблицы, владельцем которой является данный пользователь или роль и от которой зависит SQL-подпрограмма, определенная в данном каталоге
ROUTINES	Одна строка для каждой вызываемой SQL-подпрограммы в каталоге, которая доступна данному пользователю или роли
SCHEMATA	Одна строка для каждой схемы в базе данных, принадлежащей текущему пользователю; содержит имя схемы, набор символов по умолчанию и т.д.
SEQUENCES	Одна строка для каждого генератора внешней последовательности, который определен в данном каталоге и доступен данному пользователю или роли
SQL_FEATURES	Одна строка для каждой возможности или подвозможности из стандарта SQL, указывающая, реализована ли она в данной реализации SQL
SQL_IMPLEMENTATION_INFO	Одна строка для каждого определенного в стандарте SQL элемента информации реализации SQL, указывающая наличие поддержки в данной реализации SQL
SQL_LANGUAGES	Одна строка для каждого языка (например, COBOL, C и т.д.), поддерживаемого СУБД данного типа; в строке указывается уровень соответствия стандарту SQL, тип поддерживаемого диалекта SQL и т.д.
SQL_PACKAGES	Одна строка для каждого пакета стандарта SQL, указывающая, поддерживается ли этот пакет в данной реализации SQL
SQL_PARTS	Одна строка для каждой части стандарта SQL, указывающая, поддерживается ли эта часть в данной реализации SQL
SQL_SIZING	Одна строка для каждого элемента размера стандарта SQL, указывающая, поддерживается ли этот размер в данной реализации SQL
SQL_SIZING_PROFILES	Одна строка для каждого элемента размера стандарта SQL, указывающая размер, требуемый одним или несколькими профилями стандарта
TABLE_CONSTRAINTS	Одна строка для каждого ограничения (первичный ключ, внешний ключ, условие уникальности или условие на значение), заданного для таблицы, которой владеет текущий пользователь; содержит имя ограничения и таблицы, тип ограничения и его характеристики
TABLE_METHOD_PRIVILEGES	Одна строка для каждой привилегии для работы с методом, определенным над таблицами структурных типов, которые определены в каталоге и доступны (или права на них переданы) данному пользователю или роли

Представление	Содержимое
TABLE_PRIVILEGES	Одна строка для каждой привилегии для работы с таблицей, предоставленной текущему пользователю или предоставленной им другому пользователю; содержит имя таблицы, тип привилегии, указание на то, кто предоставил привилегию, кому она предоставлена и имеет ли текущий пользователь право предоставления этой привилегии
TABLES	Одна строка для каждой таблицы или каждого представления, доступных текущему пользователю; содержит имя объекта и тип (т.е. идет ли речь о таблице или представлении)
TRANSFORMS	Одна строка для каждого преобразования определенных в этом каталоге пользовательских типов, которые доступны данному пользователю или роли
TRANSLATIONS	Одна строка для каждого определения правила конвертирования текста, доступного текущему пользователю
TRIGGERED_UPDATE_COLUMNS	Одна строка для каждого столбца в каталоге, который является явным столбцом триггера события UPDATE, определенного в этом каталоге, и который доступен данному пользователю или роли
TRIGGER_COLUMN_USAGE	Одна строка для каждого столбца, от которого зависит определенный в этом каталоге триггер, владельцем которого является данный пользователь
TRIGGER_ROUTINE_USAGE	Одна строка для выполняемой SQL-подпрограммы, владельцем которой является данный пользователь или роль, от которой зависит определенный в этом каталоге триггер
TRIGGER_SEQUENCE_USAGE	Одна строка для каждого генератора внешней последовательности, владельцем которого является данный пользователь или роль и от которого зависит некоторый определенный в этом каталоге триггер
TRIGGER_TABLE_USAGE	Одна строка для каждой таблицы, от которой зависит определенный в этом каталоге триггер, владельцем которого является данный пользователь или роль
TRIGGERS	Одна строка для каждого определенного над таблицей в этом каталоге триггера, доступного данному пользователю или роли
UDT_PRIVILEGES	Одна строка для каждой привилегии для работы с пользовательскими типами в данном каталоге, которые доступны (или права на которые переданы) данному пользователю или роли
USAGE_PRIVILEGES	Одна строка для каждой привилегии использования, предоставленной текущему пользователю или переданной им другому пользователю
USER_DEFINED_TYPES	Одна строка для каждого определенного в этом каталоге пользовательского типа, который доступен данному пользователю или роли
VIEW_COLUMN_USAGE	Одна строка для каждого столбца, на который имеется ссылка в представлениях, принадлежащих текущему пользователю; содержит имя столбца и таблицы, в которую входит столбец
VIEW_ROUTINE_USAGE	Одна строка для каждой подпрограммы, владельцем которой является данный пользователь или роль и от которой зависит представление, определенное в этом каталоге
VIEW_TABLE_USAGE	Одна строка для каждой таблицы, на которую имеется ссылка в определениях представлений, принадлежащих текущему пользователю; содержит имя таблицы
VIEWS	Одна строка для каждого представления, доступного текущему пользователю; содержит имя, информацию о проверках и возможности обновления