

Vistas de catálogo del sistema

Las funciones y vistas de administración dinámica devuelven información sobre el estado del servidor que se puede utilizar para controlar el estado de una instancia del servidor, para diagnosticar problemas y para optimizar el rendimiento.

Algunas vistas de catálogo heredan filas de otras vistas de catálogo. Por ejemplo, el `sys.tables` vista de catálogo se hereda de la `sys.objects` vista de catálogo. La vista de catálogo `sys.objects` se denomina vista base y la vista `sys.tables` se denomina vista derivada. La vista de catálogo `sys.tables` devuelve las columnas específicas de tablas y todas las columnas que devuelve la vista de catálogo `sys.objects`. La vista de catálogo `sys.objects` devuelve filas de objetos distintos de tablas, como procedimientos almacenados y vistas. Después de crear una tabla, sus metadatos se devuelven en ambas vistas. Si bien las dos vistas de catálogo devuelven diferentes niveles de información sobre la tabla, solo existe una entrada en los metadatos para esta tabla con un nombre y un `object_id`. Esto se puede resumir de la manera siguiente: La vista base contiene un subconjunto de columnas y un superconjunto de filas.

La vista derivada contiene un superconjunto de columnas y un subconjunto de filas.

Las vistas de catálogo no contienen información sobre los datos de catálogo de replicación, copia de seguridad, plan de mantenimiento de bases de datos o Agente SQL Server.

Importante!

En versiones futuras de SQL Server, Microsoft puede aumentar la definición de cualquier vista de catálogo del sistema y agregar columnas al final de la lista. Se recomienda no usar la sintaxis `SELECT * FROM sys.catalog_view_name` en producción código porque el número de columnas devueltas podría cambiar y alterar la aplicación.

sys.objects

Nombre de columna	tipo de dato	descripción	valores
<code>name</code>	<code>sysname</code>	nombre del objeto	
<code>object_id</code>	<code>int</code>	Número de identificación del objeto. Es único en una base de datos	
<code>principal_id</code>	<code>int</code>	Identificador del propietario individual, si es diferente del propietario del esquema. De forma predeterminada, los objetos contenidos en el esquema pertenecen al propietario del esquema. No obstante, es posible especificar un propietario alternativo mediante la instrucción <code>ALTER AUTHORIZATION</code> para cambiar la propiedad.	Es NULL si no existe ningún propietario individual alternativo. Es NULL si el tipo de objeto es uno de los siguientes: C = <i>CHECK constraint</i> D = <i>DEFAULT (constraint or stand-alone)</i> F = <i>FOREIGN KEY constraint</i> PK = <i>PRIMARY KEY constraint</i> R = <i>Rule (old-style, stand-alone)</i> TA = <i>Assembly (CLR-integration)</i> trigger TR = <i>SQL trigger</i> UQ = <i>UNIQUE constraint</i>
<code>schema_id</code>	<code>int</code>	Identificador del objeto al que pertenece este objeto. 0 = <i>No es un objeto secundario.</i>	



By **djbarrios**
cheatography.com/djbarrios/

Not published yet.
 Last updated 30th May, 2018.
 Page 1 of 6.

Sponsored by **Readability-Score.com**
 Measure your website readability!
<https://readability-score.com>

sys.objects (cont)

type char(2) Tipo de objeto

AF = Aggregate function (CLR) **C** = CHECK constraint **D** = DEFAULT (constraint or stand-alone) **F** = FOREIGN KEY constraint **FN** = SQL scalar function **FS** = Assembly (CLR) scalar-function **FT** = Assembly (CLR) table-valued function **IF** = SQL inline table-valued function **IT** = Internal table **P** = SQL Stored Procedure **PC** = Assembly (CLR) stored-procedure **PG** = Plan guide **PK** = PRIMARY KEY constraint **R** = Rule (old-style, stand-alone) **RF** = Replication-filter-procedure **S** = System base table **SN** = Synonym **SO** = Sequence object **U** = Table (user-defined) **V** = View **Se aplica a:** desde SQL Server 2012 (11.x) hasta SQL Server 2017. **SQ** = Service queue **TA** = Assembly (CLR) DML trigger **TF** = SQL table-valued-function **TR** = SQL DML trigger **TT** = Table type **UQ** = UNIQUE constraint **X** = Extended stored procedure **Se aplica a:** SQL Server 2016 (13.x) a través de SQL Server 2017, Se aplica a: Base de datos SQL de Azure, Almacenamiento de datos SQL de Azure, Almacenamiento de datos paralelos. **ET** = External Table

C

By **djbarrios**
cheatography.com/djbarrios/

Not published yet.
Last updated 30th May, 2018.
Page 2 of 6.

Sponsored by **Readability-Score.com**
Measure your website readability!
<https://readability-score.com>

sys.objects (cont)

type_desc	nvarchar (60)	Descripción del tipo de objeto	AGGREGATE_FUNCTION CHECK_CONSTRAINT CLR_SCALAR_FUNCTION CLR_STORED_PROCEDURE CLR_TABLE_VALUED_FUNCTION CLR_TRIGGER DEFAULT_CONSTRAINT EXTENDED_STORED_PROCEDURE FOREIGN_KEY_CONSTRAINT INTERNAL_TABLE PLAN_GUIDE PRIMARY_KEY_CONSTRAINT REPLICATION_FILTER_PROCEDURE RULE SEQUENCE_OBJECT Se aplica a: desde SQL Server 2012 (11.x) hasta SQL Server 2017. SERVICE_QUEUE SQL_INLINE_TABLE_VALUED_FUNCTION SQL_SCALAR_FUNCTION SQL_STORED_PROCEDURE SQL_TABLE_VALUED_FUNCTION SQL_TRIGGER SYNONYM SYSTEM_TABLE TABLE_TYPE UNIQUE_CONSTRAINT USER_TABLE VIEW
-----------	---------------	--------------------------------	--

create_date datetime Fecha de creación del objeto

modify_date datetime Fecha en que se modificó el objeto por última vez con una instrucción ALTER. Si el objeto es una tabla o una vista, modify_date también cambia cuando se crea o modifica un índice clúster en la tabla o la vista.

is_ms_shipped bit Un componente interno de SQL Server creó el objeto.

is_published bit El objeto se publica.

is_schema_published bit Solo se ha publicado el esquema del objeto.

Contiene una fila por cada objeto definido por el usuario, el ámbito de esquema que se crea dentro de una base de datos, incluidos los compilados de forma nativa función escalar definida por el usuario.

sys.objects no muestra los desencadenadores DDL, porque no tienen el ámbito de esquema. Todos los desencadenadores, tanto DML como DDL, se encuentran en sys.triggers. Sys.Triggers admite una mezcla de reglas de ámbito de nombre para los distintos tipos de desencadenadores.

Comentarios sys.objects

Puede aplicar el OBJECT_ID, OBJECT_NAME, y OBJECTPROPERTY las funciones integradas de () para los objetos mostrados en sys.objects. Hay una versión de esta vista con el mismo esquema, denominado sys.system_objects, que muestra los objetos del sistema. Hay otra vista denominada sys.all_objects que muestra los objetos de usuario y del sistema. Las tres vistas de catálogo tienen la misma estructura. En esta versión de SQL Server, un índice extendido, por ejemplo un índice XML o espacial, se considera como una tabla interna en sys.objects (type = IT y type_desc = INTERNAL_TABLE). En un índice extendido: name es el nombre interno de la tabla de índice. parent_object_id es el object_id de la tabla base. Las columnas is_ms_shipped, is_published y is_schema_published están establecidas en 0.



sys.columns			
Nombre de columna	tipo de dato	descripción	valores
object_id	int	Identificador del objeto al que pertenece esta columna.	
name	sysname	Nombre de la columna. Es único en el objeto.	
column_id	int	Identificador de la columna. Es único en el objeto. Los Id. de columna no tienen que ser secuenciales.	
system_type_id	tinyint	Id. del tipo de sistema de la columna.	
user_type_id	int	Id. del tipo de la columna, tal como lo ha definido el usuario. Para devolver el nombre del tipo, unir a la sys.types vista en esta columna de catálogo.	
max_length	smallint	Longitud máxima de la columna, en bytes.	-1 = la columna es de tipo de dato <i>varchar (max)</i> , <i>nvarchar (max)</i> , <i>varbinary (max)</i> , o <i>xml</i> . Para columnas de tipo text, el valor max_length será 16 o el valor establecido por la opción "text in row" de sp_tableoption.
precision	tinyint	Precisión de la columna si es basada en números; en caso contrario, es 0.	
escala	tinyint	La escala de la columna se basa en valores numéricos; en caso contrario, es 0.	
collation_name	sysname	Nombre de la intercalación de la columna si basados en caracteres; en caso contrario, es NULL.	
is_nullable	bit		1 = La columna acepta valores NULL.
is_ansi_padded	bit		1 = La columna utiliza el comportamiento ANSI_PADDING ON si es de tipo character, binary o variant. 0 = La columna no es de tipo character, binary o variant.



By **djbarrios**
cheatography.com/djbarrios/

Not published yet.
 Last updated 30th May, 2018.
 Page 4 of 6.

Sponsored by **Readability-Score.com**
 Measure your website readability!
<https://readability-score.com>

sys.columns (cont)

is_rowguidcol	bit		1 = La columna se ha declarado como ROWGUIDCOL.
is_identity	bit		1 = La columna tiene valores de identity.
is_computed	bit		1 = La columna es una columna calculada.
is_filestream	bit		1 = La columna es una columna FILESTREAM.
is_replicated	bit		1 = La columna está replicada.
is_non_sql_subscribed	bit		1 = La columna tiene un suscriptor que no es de SQL Server.
is_merge_published	bit		= La columna es merge-published
is_dts_replicated	bit		1 = La columna se replica con SSIS.
is_xml_document	bit		1 = El contenido es un documento XML completo. 0 = el contenido es un fragmento de documento o el tipo de datos de columna no es xml.
xml_collation_id	int	Es distinto de cero si el tipo de datos de la columna es xml y se ha escrito el código XML. El valor será el identificador de la colección que contiene el espacio de nombres de esquema XML validación de la columna.	0 = No es una colección de esquemas XML.
default_object_id	int	Id. del objeto predeterminado, independientemente de si es un objeto independiente sys.sp_bindefault, o una restricción de valor predeterminado de nivel de columna insertada. La columna parent_object_id de un objeto predeterminado de nivel de columna insertada es una referencia a la propia tabla.	0 = No hay un valor predeterminado.
rule_object_id	int	Id. de la regla independiente enlazada a la columna mediante sys.sp_bindrule.	0 = No hay ninguna regla independiente. Para las restricciones CHECK de nivel de columna, vea sys.check_constraints (Transact-SQL).



sys.columns (cont)

is_sparse	bit		1 = La columna es una columna dispersa. Para obtener más información, vea Usar columnas dispersas.
is_column_set	bit		1 = La columna es un conjunto de columnas. Para obtener más información, vea Usar columnas dispersas.
generated_always_type	tinyint	Se aplica a: de SQL Server 2016 (13.x) a SQL Server 2017, Base de datos SQL. Identifica cuando se genera el valor de columna (siempre será 0 para las columnas en tablas del sistema)	0 = NOT_APPLICABLE 1 = AS_ROW_START 2 = AS_ROW_END Para obtener más información, consulte tablas temporales (bases de datos relacionales).
generated_always_type_desc	nvarchar(60)	Se aplica a: de SQL Server 2016 (13.x) a SQL Server 2017, Base de datos SQL. Descripción textual del generated_always_type del valor (siempre NOT_APPLICABLE para las columnas en tablas del sistema)	NOT_APPLICABLE AS_ROW_START AS_ROW_END

Devuelve una fila para cada columna de un objeto que incluye columnas, como vistas o tablas. La siguiente lista incluye tipos de objetos que contienen columnas:

Table-valued assembly functions (**FT**)

Inline table-valued SQL functions (**IF**)

Internal tables (**IT**)

System tables (**S**)

Table-valued SQL functions (**TF**)

User tables (**U**)

Views (**V**)

