

# CHAPTER 16

## System Views

### About this chapter

This chapter lists predefined views for the system tables.

The system tables described in "System Tables" on page 771 use numbers to identify tables, user IDs, and so forth. While this is efficient for internal use, it makes these tables difficult for people to interpret. A number of predefined system views are provided that present the information in the system tables in a more readable format.

The definitions for the system views are included with their descriptions. Some of these definitions are complicated, but need not be understood to use the views. They serve as good examples of what can be accomplished using the SELECT command and views.

### Contents

<b>Topic</b>	<b>Page</b>
SYSARTICLECOLS system view	829
SYSARTICLES system view	830
SYSCATALOG system view	831
SYSCOLAUTH system view	832
SYSCOLUMNS system view	833
SYSFOREIGNKEYS system view	834
SYSGROUPS system view	835
SYSINDEXES system view	836
SYSOPTIONS system view	837
SYSROCAUTH system view	838
SYSROCPARMS system view	839
SYSROCPUBLICATIONS system view	840
SYSROCMOTETYPES system views	841
SYSROCMOTEUSERS system view	842
SYSROCSUBSCRIPTIONS system view	843
SYSROCTABAUTH system view	844

SYSTRIGGERS system view	845
SYSUSERAUTH system view	846
SYSUSERLIST system view	847
SYSUSEROPTIONS system view	848
SYSUSERPERMS system view	849
SYSVIEWS system view	850
Views for Transact-SQL Compatibility	851

## **SYSARTICLECOLS system view**

```
CREATE VIEW SYS.SYSARTICLECOLS
AS SELECT (select publication_name FROM
SYS.SYSPUBLICATION AS p
WHERE p.publication_id=ac.publication_id) AS
publication_name,
(select table_name FROM SYS.SYSTABLE AS t
WHERE t.table_id=ac.table_id) AS table_name,
select column_name FROM SYS.SYSCOLUMN AS c
WHERE c.table_id=ac.table_id
AND c.column_id=ac.column_id) AS column_name
FROM SYS.SYSARTICLECOL AS ac
```

Presents a readable version of the table **SYSARTICLECOLS**.

## **SYSARTICLES system view**

```
CREATE VIEW SYS.SYSARTICLES
AS SELECT(select publication_name FROM
SYS.SYSPUBLICATION AS p
WHERE p.publication_id=a.publication_id) AS
publication_name,
(select table_name FROM SYS.SYSTABLE AS t
WHERE t.table_id=a.table_id) AS table_name,
where_expr,subscribe_by_expr
FROM SYS.SYSARTICLE AS a
```

Presents a readable version of the table **SYSARTICLES**.

## SYSCATALOG system view

```
CREATE VIEW SYS.SYSCATALOG ( creator,
                             tname, dbspacename, tabletype, ncols,
                             primary_key, "check", remarks )
AS
SELECT ( SELECT user_name FROM SYS.SYSUSERPERM
          WHERE user_id = SYSTABLE.creator ),
       table_name,
       ( SELECT dbspace_name from SYS.SYSFILE
         WHERE file_id = SYSTABLE.file_id ),
       IF table_type='BASE' THEN 'TABLE'
       ELSE table_type ENDIF,
       ( SELECT count(*) FROM SYS.SYSCOLUMN
         WHERE table_id = SYSTABLE.table_id ),
       IF primary_root = 0 THEN 'N' ELSE 'Y' ENDIF,
       IF table_type <> VIEW' THEN view_def ENDIF,
       remarks
FROM SYS.SYSTABLE
```

Lists all the tables and views from **SYSTABLE** in a readable format.

## **SYSCOLAUTH system view**

```
CREATE VIEW SYS.SYSCOLAUTH ( grantor, grantee,
                             creator, tname, colname )
AS
SELECT ( SELECT user_name FROM SYS.SYSUSERPERM
          WHERE user_id = SYSCOLPERM.grantor ),
       ( SELECT user_name FROM SYS.SYSUSERPERM
          WHERE user_id = SYSCOLPERM.grantee ),
       ( SELECT user_name
         FROM SYS.SYSUSERPERM == SYS.SYSTABLE
         WHERE table_id = SYSCOLPERM.table_id ),
       ( SELECT table_name FROM SYS.SYSTABLE
         WHERE table_id = SYSCOLPERM.table_id ),
       ( SELECT column_name FROM SYS.SYSCOLUMN
         WHERE table_id = SYSCOLPERM.table_id
         AND column_id = SYSCOLPERM.column_id )
FROM SYS.SYSCOLPERM
```

Presents column update permission information in **SYSCOLPERM** in a more readable format.

## SYSCOLUMNS system view

```
CREATE VIEW SYS.SYSCOLUMNS ( creator, cname, tname,
coltype, nulls, length, syslength,
in_primary_key, "colno", default_value, remarks )
AS
SELECT ( SELECT user_name FROM SYS.SYSUSERPERM
        WHERE user_id = SYSTABLE.creator ),
       column_name, table_name,
       ( SELECT domain_name FROM SYS.SYSDOMAIN
         WHERE domain_id = SYSCOLUMN.domain_id ),
       nulls, width, scale, pkey, column_id,
       "default", SYSCOLUMN.remarks
FROM SYS.SYSCOLUMN == SYS.SYSTABLE
```

Presents a readable version of the table **SYSCOLUMN**. (Note the S at the end of the view name that distinguishes it from the **SYSCOLUMN** table.)

## SYSFOREIGNKEYS system view

```
CREATE VIEW SYS.SYSFOREIGNKEYS ( foreign_creator,  
    foreign_tname, primary_creator,  
    primary_tname, role, columns )  
AS  
SELECT ( SELECT user_name FROM  
    SYS.SYSUSERPERM == SYS.SYSTABLE  
    WHERE table_id = foreign_table_id ),  
( SELECT table_name FROM SYS.SYSTABLE  
    WHERE table_id = foreign_table_id ),  
( SELECT user_name  
    FROM SYS.SYSUSERPERM == SYS.SYSTABLE  
    WHERE table_id = primary_table_id ),  
( SELECT table_name FROM SYS.SYSTABLE  
    WHERE table_id = primary_table_id ), role,  
( SELECT list( string( FK.column_name,  
    ' IS ', PK.column_name ) )  
    FROM SYS.SYSFKCOL KEY JOIN  
    SYS.SYSCOLUMN FK, SYS.SYSCOLUMN PK  
    WHERE foreign_table_id =  
    SYSFOREIGNKEY.foreign_table_id  
    AND foreign_key_id =  
    SYSFOREIGNKEY.foreign_key_id  
    AND PK.table_id =  
    SYSFOREIGNKEY.primary_table_id  
    AND PK.column_id =  
    SYSFKCOL.primary_column_id )  
FROM SYS.SYSFOREIGNKEY
```

Presents foreign key information from **SYSFOREIGNKEY** and **SYSFKCOL** in a more readable format.



## **SYSGROUPS system view**

```
CREATE VIEW SYS.SYSGROUPS ( group_name, member_name )
AS
SELECT g.user_name, u.user_name
FROM   SYS.SYSGROUP,
       SYS.SYSUSERPERM g,
       SYS.SYSUSERPERM u
WHERE  group_id = g.user_id
AND    group_member = u.user_id
```

Presents group information from SYSGROUP in a more readable format.

## SYSINDEXES system view

```
CREATE VIEW SYS.SYSINDEXES ( icreator, iname, fname,
creator,
tname, indextype, colnames, interval, level )
AS
SELECT ( SELECT user_name FROM SYS.SYSUSERPERM
WHERE user_id = SYSINDEX.creator ),
index_name,
( SELECT file_name FROM SYS.SYSFILE
WHERE file_id = SYSINDEX.file_id ),
( SELECT user_name FROM SYS.SYSUSERPERM
WHERE user_id = SYSINDEX.creator ),
table_name,
IF "unique" = 'Y' THEN 'Unique'
ELSE 'Non-unique' ENDIF,
( SELECT list( string( column_name,
IF "order" = 'A' THEN ' ASC' i
ELSE ' DESC' ENDIF ) )
FROM SYS.SYSIXCOL == SYS.SYSCOLUMN
WHERE index_id = SYSINDEX.index_id ), 0, 0
FROM SYS.SYSTABLE KEY JOIN SYS.SYSINDEX
```

Presents index information from **SYSINDEX** and **SYSIXCOL** in a more readable format.

## **SYSOPTIONS system view**

```
CREATE VIEW SYS.SYSOPTIONS ( user_name, "option",  
"setting" )  
AS  
SELECT ( SELECT user_name FROM SYS.SYSUSERPERM  
        WHERE user_id = SYSOPTION.user_id ),  
        "option", "setting"  
FROM SYS.SYSOPTION
```

Presents option settings contained in the table **SYSOPTION** in a more readable format.

## **SYSPROCAUTH system view**

```
CREATE VIEW SYS.SYSPROCAUTH(grantee,  
    creator,procname)  
AS  
SELECT(select user_name FROM SYS.SYSUSERPERM  
    WHERE SYSPROCPERM.grantee=SYSUSERPERM.user_id),  
    (select user_name FROM SYS.SYSUSERPERM  
    WHERE SYSPROCEDURE.creator=SYSUSERPERM.user_id),  
    proc_name  
FROM  
SYS.SYSPROCEDURE JOIN SYS.SYSPROCPERM
```

Presents the procedure authorities from **SYSUSERPERM** in a more readable format.

## SYSPROCPARMS system view

```
CREATE VIEW SYS.SYSPROCPARMS ( creator, parmname,  
procname,  
parmtype, parmmode, parmdomain, length, remarks )  
AS  
SELECT ( SELECT user_name FROM SYS.SYSUSERPERM  
        WHERE user_id = SYSPROCEDURE.creator ),  
parm_name, proc_name, parm_type,  
IF parm_mode_in = 'Y' AND  
  parm_mode_out = 'N' THEN 'IN'  
ELSE IF parm_mode_in = 'N'  
  AND parm_mode_out = 'Y' THEN 'OUT'  
ELSE 'INOUT' ENDIF ENDIF,  
( SELECT domain_name FROM SYS.SYSDOMAIN  
  WHERE domain_id = SYSPROC.PARM.domain_id ),  
width, SYSPROC.PARM.remarks  
FROM SYS.SYSPROC.PARM == SYS.SYSPROCEDURE
```

Presents the procedure parameters from **SYSPROC.PARM** in a more readable format.

## **SYSPUBLICATIONS system view**

```
CREATE VIEW SYS.SYSPUBLICATIONS
AS
SELECT(select user_name FROM SYS.SYSUSERPERM AS u
       WHERE u.user_id=p.creator) AS creator,
       publication_name,remarks
FROM SYS.SYSPUBLICATION AS p
```

Presents the user name from the SYSUSERPERM table for all creators and displays the publication name and remarks from the SYSPUBLICATION table in a more readable format.

## **SYSREMOTETYPES system views**

```
CREATE VIEW SYS.SYSREMOTETYPES
AS SELECT type_id,type_name,publisher_address,remarks
FROM SYS.SYSREMOTETYPE
```

Presents the procedure remote types from the **SYSREMOTETYPES** in a more readable format.

## SYSREMOTEOUSERS system view

```

CREATE VIEW SYS.SYSREMOTEOUSERS
AS SELECT(SELECT user_name FROM SYS.SYSUSERPERM AS u
  WHERE u.user_id=r.user_id) AS user_name,
  "consolidate",
  (SELECT type_name FROM SYS.SYSREMOTETYPE AS t
  WHERE t.type_id=r.type_id) AS type_name,
  "address",frequency,send_time,
  (IF frequency='A' THEN
  NULL
  ELSE
  IF frequency='P' THEN
  IF time_sent IS NULL THEN
  current timestamp
  ELSE
  (SELECT min(minutes(time_sent,
  60*hour(a.send_time)
  +minute(seconds(a.send_time,59))))
  FROM SYS.SYSREMOTEOUSER AS a
  WHERE a.frequency='P'
  AND a.send_time=r.send_time)
  ENDIF
  ELSE
  IF current date+send_time
  >COALESCE(time_sent,current timestamp) THEN
  current date+send_time
  ELSE
  current date+send_time+1
  ENDIF
  ENDIF
  ENDIF) AS next_send,
  log_send,time_sent,log_sent,
  confirm_send,send_count,resent_count,
  time_received,log_received,confirm_received,
  receive_count,rereceive_count
FROM SYS.SYSREMOTEOUSER AS r

```

Presents the information from **SYSREMOTEOUSER** in a more readable format.



## SYSSUBSCRIPTIONS system view

```
CREATE VIEW SYS.SYSSUBSCRIPTIONS
AS
SELECT(select publication_name
        FROM SYS.SYSPUBLICATION AS p
        WHERE p.publication_id=s.publication_id) AS
        publication_name,
        (select user_name FROM SYS.SYSUSERPERM AS u
        WHERE u.user_id=s.user_id) AS user_name,
        subscribe_by,created,started
FROM SYS.SYSSUBSCRIPTION AS s
```

Presents subscription information, such as the publication name, creation time, and start time from the SYSPUBLICATION table in a more readable format.

## SYSTABAUTH system view

```
CREATE VIEW SYS.SYSTABAUTH ( grantor, grantee,
                             screator, stname, tcreator, tname,
                             selectauth, insertauth, deleteauth,
                             updateauth, updatecols, alterauth, referenceauth )
AS
SELECT ( SELECT user_name FROM SYS.SYSUSERPERM
          WHERE user_id = SYSTABLEPERM.grantor ),
       ( SELECT user_name FROM SYS.SYSUSERPERM
          WHERE user_id = SYSTABLEPERM.grantee ),
       ( SELECT user_name
         FROM SYS.SYSUSERPERM == SYS.SYSTABLE
         WHERE table_id = SYSTABLEPERM.stable_id ),
       ( SELECT table_name FROM SYS.SYSTABLE
         WHERE table_id = SYSTABLEPERM.stable_id ),
       ( SELECT user_name FROM
         SYS.SYSUSERPERM == SYS.SYSTABLE
         WHERE table_id = SYSTABLEPERM.ttable_id ),
       ( SELECT table_name FROM SYS.SYSTABLE
         WHERE table_id = SYSTABLEPERM.ttable_id ),
       selectauth, insertauth, deleteauth,
       updateauth, updatecols,
       alterauth, referenceauth
FROM SYS.SYSTABLEPERM
```

Presents table permission information from **SYSTABLEPERM** in a more readable format.

## SYSTRIGGERS system view

```
CREATE VIEW SYS.SYSTRIGGERS ( owner, trigname, tname,  
event, trigtime, trigdefn )  
AS  
SELECT ( SELECT user_name FROM SYS.SYSUSERPERM  
        WHERE user_id = SYSTABLE.creator ),  
        trigger_name, table_name,  
        IF event = 'I' THEN 'INSERT'  
        ELSE IF event = 'U' THEN 'UPDATE'  
        ELSE IF event = 'C' THEN 'UPDATE'  
        ELSE 'DELETE' ENDIF ENDIF ENDIF,  
        IF trigger_time = 'B' THEN 'BEFORE'  
        ELSE 'AFTER' ENDIF,  
        trigger_defn  
FROM SYS.SYSTRIGGER == SYS.SYSTABLE  
WHERE foreign_table_id IS NULL
```

Lists all the triggers from **SYSTRIGGER** in a readable format.

## **SYSUSERAUTH system view**

```
CREATE VIEW SYS.SYSUSERAUTH ( name, password,  
    resourceauth, dbaauth, scheduleauth, user_group )  
AS  
SELECT user_name, password, resourceauth,  
    dbaauth, scheduleauth, user_group  
FROM SYS.SYSUSERPERM
```

Presents all the information in the table **SYSUSERPERM** except for user numbers. Because this view shows passwords, this system view does not have **PUBLIC** select permission. (All other system views have **PUBLIC** select permission.)

## **SYSUSERLIST system view**

```
CREATE VIEW SYS.SYSUSERLIST ( name, resourceauth,  
                             dbaauth, scheduleauth, user_group )  
AS  
SELECT user_name, resourceauth,  
       dbaauth, scheduleauth, user_group  
FROM SYS.SYSUSERPERM
```

Presents all of the information in **SYSUSERAUTH** except passwords.

## **SYSUSEROPTIONS system view**

```
CREATE VIEW SYS.SYSUSEROPTIONS ( "user_name",  
    "option", "setting" )  
AS  
SELECT u.name, "option",  
    isnull( ( SELECT "setting"  
        FROM sys.sysoptions s  
        WHERE s.user_name = u.name  
        AND s."option" = o."option" ),  
    "setting" )  
FROM SYS.SYSOPTIONS o, SYS.SYSUSERAUTH u  
WHERE o.user_name = 'PUBLIC'
```

Presents permanent option settings that are in effect for each user. If a user has no setting for an option, this view displays the public setting for the option.

## **SYSUSERPERMS system view**

```
CREATE VIEW SYS.SYSUSERPERMS
AS
SELECT user_id, user_name, resourceauth, dbaauth,
       scheduleauth, user_group, remarks
FROM SYS.SYSUSERPERM
```

Contains exactly the same information as the table SYSUSERPERM, except the password is omitted. All users have read access to this view, but only the DBA has access to the underlying table (SYSUSERPERM).

## **SYSVIEWS system view**

```
CREATE VIEW SYS.SYSVIEWS ( vcreator, viewname, viewtext
)
AS
SELECT user_name, table_name, view_def
FROM SYS.SYSTABLE KEY JOIN SYS.SYSUSERPERM
WHERE table_type = 'VIEW'
```

Lists views along with their definitions.



## Views for Transact-SQL Compatibility

Adaptive Server Enterprise and Adaptive Server Anywhere have different system catalogs, reflecting the different uses for the two products.

In Adaptive Server Enterprise, a single master database contains a set of system tables, which information that applies to all databases on the server. Many databases may exist within the master database, and each has additional system tables associated with it.

In Adaptive Server Anywhere, each database exists independently, and contains its own system tables. There is no master database that contains system information on a collection of databases. Each server may run several databases at a time, dynamically loading and unloading each database as needed.

The Adaptive Server Enterprise and Adaptive Server Anywhere system catalogs are different. The Adaptive Server Enterprise system tables and views are owned by the special user *dbo*, and exist partly in the master database, partly in the **sybsecurity** database, and partly in each individual database; the Adaptive Server Anywhere system tables and views are owned by the special user SYS and exist separately in each database.

To assist in preparing compatible applications, Adaptive Server Anywhere provides a set of views owned by the special user *dbo*, which correspond to the Adaptive Server Enterprise system tables and views. Where architectural differences make the contents of a particular Adaptive Server Enterprise table or view meaningless in a Adaptive Server Anywhere context, the view is empty, containing just the column names and data types.

The following tables list the Adaptive Server Enterprise system tables and their implementation in the Adaptive Server Anywhere system catalog. The owner of all tables is *dbo* in each DBMS.

Tables existing in each Adaptive Server Enterprise database

Table name	Description	Data?
sysalternates	One row for each user mapped to a database user	No
syscolumns	One row for each column in a table or view, and for each parameter in a procedure	Yes
syscomments	One or more rows for each view, rule, default, trigger, and procedure, giving the SQL definition statement	Yes
sysconstraints	One row for each referential or check constraint associated with a table or column	No
sysdepends	One row for each procedure, view, or table that is referenced by a procedure, view, or trigger	No

Table name	Description	Data?
sysindexes	One row for each clustered or nonclustered index, one row for each table with no indexes, and an additional row for each table containing text or image data.	Yes
syskeys	One row for each primary, foreign, or common key; set by the user (not maintained by Adaptive Server Enterprise)	No
syslogs	Transaction log	No
sysobjects	One row for each table, view, procedure, rule, trigger default, log, or (in tempdb only) temporary object	Contains compatible data only
sysprocedures	One row for each view, rule, default, trigger, or procedure, giving the internal definition	No
sysprotects	User permissions information	No
sysreferences	One row for each referential integrity constraint declared on a table or column	No
sysroles	Maps server-wide roles to local database groups	No
syssegments	One row for each segment (named collection of disk pieces)	No
systhresholds	One row for each threshold defined for the database	No
systypes	One row for each system-supplied or user-defined data type	Yes
sysusermessages	One row for each user-defined message	Yes (this is an Adaptive Server Anywhere system table)
sysusers	One row for each user allowed in the database	Yes

Tables existing in the Adaptive Server Enterprise master database

Table name	Description	Data?
syscharsets	One row for each character set or sort order	No
sysconfigures	One row for each user-settable configuration parameter	No
syscurconfigs	Information about configuration parameters currently being used by the server	No
sysdatabases	One row for each database on the server	No

Table name	Description	Data?
sysdevices	One row for each tape dump device, disk dump device, disk for databases, or disk partition for databases	No
sysengines	One row for each server currently online	No
syslanguages	One row for each language (except U.S. English) known to the server	No
syslocks	Information about active locks	No
sysloginroles	One row for each server login that possesses a system-defined role	No
syslogins	One row for each valid user account	Yes
sysmessages	One row for each system error or warning	No
sysprocesses	Information about server processes	No
sysremotelogins	One row for each remote user	No
sysrvroles	One row for each server-wide role	No
syssservers	One row for each remote server	No
sysusages	One row for each disk piece allocated to a database	No

Tables existing in the Adaptive Server Enterprise sybsecurity database

Table name	Description	Data?
sysaudits	One row for each audit record	No
sysauditoptions	One row for each global audit option	No

