

# System Tables

## About this chapter

The structure of every database is described in a number of **system tables**.

The Entity-Relationship diagram on the next page shows all the system tables and the foreign keys that connect them.

The system tables are owned by the **SYS** user ID. The contents of these tables can be changed only by the database system. The UPDATE, DELETE, and INSERT commands cannot be used to modify the contents of these tables. Further, the structure of these tables cannot be changed using the ALTER TABLE and DROP commands.

## Contents

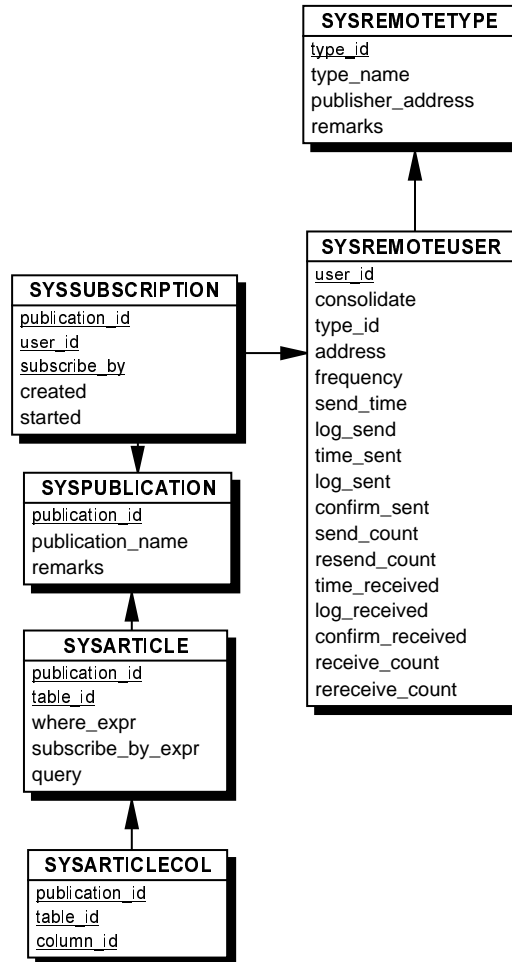
Topic	Page
System tables diagram	773
System table descriptions	776
DUMMY system table	777
SYSARTICLE system table	778
SYSARTICLECOL system table	779
SYSCOLLATION system table	780
SYSCOLLATIONMAPPINGS system table	781
SYSCOLPERM system table	782
SYSCOLUMN system table	783
SYSDOMAIN system table	785
SYSEXTENT system table	786
SYSEXTERNLOGINS system table	787
SYSFILE system table	788
SYSFKCOL system table	789
SYSFOREIGNKEY system table	790
SYSGROUP system table	792
SYSINDEX system table	793
SYSINFO system table	794

SYSIXCOL system table	796
SYSJAR system table	797
SYSJARCOMPONENT system table	798
SYSJAVACLASS system table	799
SYSLOGIN system table	801
SYSOPTION system table	802
SYSPROCEDURE system table	803
SYSROCPARM system table	804
SYSROCPERM system table	806
SYSPUBLICATIN system table	807
SYSREMOTETYPE system table	808
SYSREMOTEUSER system table	809
SYSSEVERERS system table	811
SYSSEQLSERVERTYPE system table	812
SYSSEBSUBSCRIPTION system table	813
SYSTABLE system table	814
SYSTABLEPERM system table	816
SYSSTRIGGER system table	818
SYSTYPEMAP system table	820
SYSUSERMESSAGES system table	821
SYSUSERPERM system table	822
SYSUSERTYPE system table	824



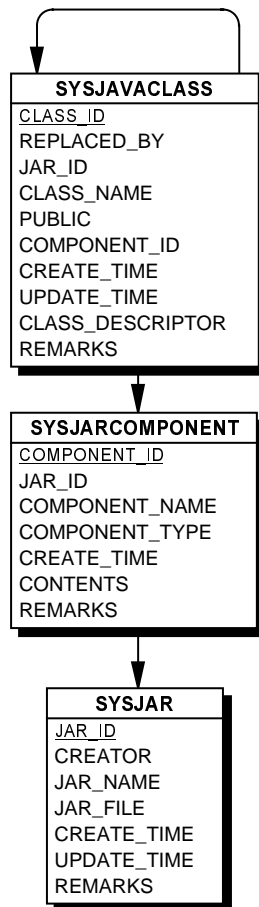
## SQL Remote system tables

The system tables used for SQL Remote are listed below. Foreign key relations between tables are indicated by arrows: the arrow leads from the foreign table to the primary table.



## Java system tables

Those system tables that are used for Java in the database are listed below. Foreign key relations between tables are indicated by arrows: the arrow leads from the foreign table to the primary table.



## **System table descriptions**

This chapter contains descriptions of each of the system tables. Several of the columns have only two possible values. Usually these values are "Y" and "N" for "yes" and "no" respectively. These columns are designated by "(Y/N)".

## DUMMY system table

Column name	Column type	Column constraint	Table constraints
dummy_col	INTEGER	NOT NULL	

The **DUMMY** table is provided as a table that always has exactly one row. This can be useful for extracting information from the database, as in the following example that gets the current user ID and the current date from the database.

```
SELECT USER, today(*) FROM SYS.DUMMY
```

**dummy\_col** This column is not used. It is present because a table cannot be created with no columns.

## SYSARTICLE system table

Column name	Column type	Column constraint	Table constraints
publication_id	SMALLINT	NOT NULL	Primary key
table_id	SMALLINT	NOT NULL	Primary key
where_expr	LONG VARCHAR		
subscribe_by_expr	LONG VARCHAR		

Each row of **SYSARTICLE** describes an article in a SQL Remote publication.

**publication\_id** The publication of which this article is a part.

**table\_id** Each article consists of columns and rows from a single table. This column contains the table ID for this table.

**where\_expr** For articles that contain a subset of rows defined by a WHERE clause, this column contains the search condition.

**subscribe\_by\_expr** For articles that contain a subset of rows defined by a SUBSCRIBE BY expression, this column contains the expression.



## SYSARTICLECOL system table

Column name	Column type	Column constraint	Table constraints
publication_id	SMALLINT	NOT NULL	Primary Key, foreign key references SYS.SYSARTICLE
table_id	SMALLINT	NOT NULL	Primary Key, foreign key references SYS.SYSARTICLE, SYS.SYSCOLUMN
column_id	SMALLINT	NOT NULL	Primary Key, foreign key references SYS.SYSCOLUMN

Each row identifies a column in an article.

**publication\_id** A unique identifier for the publication of which the column is a part.

**table\_id** The table to which the column belongs.

**column\_id** The column identifier, from the SYSCOLUMN system table.

## SYSCOLLATION system table

Column name	Column type	Column constraint	Table constraint
collation_id	SMALLINT	NOT NULL	Primary key
collation_label	CHAR(10)	NOT NULL	
collation_name	CHAR(128)	NOT NULL	
collation_order	BINARY(1280)	NOT NULL	

This table contains the collation sequences available to Adaptive Server Anywhere. There is no way to modify the contents of this table.

**collation\_id** A unique number identifying the collation sequence. The collation sequence with **collation\_id** 2 is the sequence used in previous versions of Adaptive Server Anywhere, and is the default when a database is created.

**collation\_label** A string identifying the collation sequence. The collation sequence to be used is selected when the database is created, by specifying the collation label with the `-z` option.

**collation\_name** The name of the collation sequence.

**collation\_order** An array of bytes defining how each of the 256 character codes are treated for comparison purposes. All string comparisons translate each character according to the collation order table before comparing the characters. For the different ASCII code pages, the only difference is how accented characters are sorted. In general, an accented character is sorted as if it were the same as the nonaccented character.

## SYSCOLLATIONMAPPINGS system table

Column name	Column type	Column constraint	Table Constraints
collation_label	char(10)	NOT NULL	Primary key
collation_name	char(128)	NOT NULL	
cs_label	char(128)		
so_case_label	char(128)		
so_caseless_label	char(128)		
jdk_label	char(128)		

**collation\_label** A string identifying the collation sequence. The collation sequence to be used is selected when the database is created, by specifying the collation label with the `-z` option.

**collation\_name** The collation name used to describe the character set encoding.

**cs\_label** The GPG character set mapping label.

**so\_case\_label** The collation sort order for case-sensitive GPG character set mapping.

**so\_caseless\_label** The collation sort order for case-insensitive GPG character set mapping.

**jdk\_label** The JDK character set label.

## SYSCOLPERM system table

Column name	Column type	Column constraint	Table constraint
table_id	SMALLINT	NOT NULL	Primary key, foreign key references SYS.SYSCOLUMN
grantee	SMALLINT	NOT NULL	Primary key, foreign key references SYS.SYSUSERPERM
grantor	SMALLINT	NOT NULL	Primary key, foreign key references SYS.SYSUSERPERM.user_id
column_id	SMALLINT	NOT NULL	Primary key, foreign key references SYS.SYSCOLUMN
privilege_type	SMALLINT	NOT NULL	Primary key
is_grantable	CHAR(1)	NOT NULL	

The GRANT command can give UPDATE permission to individual columns in a table. Each column with UPDATE permission is recorded in one row of **SYSCOLPERM**.

**table\_id** The table number for the table containing the column.

**grantee** The user number of the user ID that is given UPDATE permission on the column. If the **grantee** is the user number for the special **PUBLIC** user ID, the UPDATE permission is given to all user IDs.

**grantor** The user number of the user ID that grants the permission.

**column\_id** This column number, together with the **table\_id**, identifies the column for which UPDATE permission has been granted.

**privilege\_type** The number in this column indicates the kind of column permission (REFERENCES, SELECT or UPDATE).

**is\_grantable (Y/N)** Indicates if the permission on the column was granted WITH GRANT OPTION.

## SYSCOLUMN system table

Column name	Column type	Column constraint	Table constraints
table_id	SMALLINT	NOT NULL	Primary key, foreign key references SYS.SYSTABLE. table_id
column_id	SMALLINT	NOT NULL	Primary key, foreign key references SYS.SYSIXCOL
pkey	CHAR(1)	NOT NULL	
domain_id	SMALLINT	NOT NULL	foreign key references SYS.SYSDOMAI N.domain_id
nulls	CHAR(1)	NOT NULL	
width	SMALLINT	NOT NULL	
scale	SMALLINT	NOT NULL	
estimate	INTEGER	NOT NULL	
column_name	CHAR(128)	NOT NULL	
remarks	LONG VARCHAR		
"default"	LONG VARCHAR		
"check"	LONG VARCHAR		
user_type	SMALLINT		Foreign key references SYS.SYSUSERT YPE. type_id
format_str	CHAR(128)		
column_type	CHAR(1)	NOT NULL	
remote_name	VARCHAR(128)		
remote_type	SMALLINT		

Each column in every table or view is described by one row in **SYSCOLUMN**.

**table\_id** A number that uniquely identifies the table or view to which this column belongs.

**column\_id** Each table starts numbering columns at 1. The order of column numbers determines the order that columns are displayed in the command

```
SELECT * FROM TABLE
```

**pkey (Y/N)** Indicate whether this column is part of the primary key for the table.

**domain\_id** The data type for the column, indicated by a data type number listed in the **SYSDOMAIN** table.

**nulls (Y/N)** Indicates whether the NULL value is allowed in this column.

**width** The length of a string column, the precision of numeric columns or the number of bytes of storage for any other data type.

**scale** The number of digits after the decimal point for numeric data type columns, and zero for all other data types.

**estimate** A self-tuning parameter for the optimizer. Adaptive Server Anywhere will learn from previous queries by adjusting guesses that are made by the optimizer.

**column\_name** The name of the column.

**remarks** A comment string.

**default** The default value for the column. This value is only used when an INSERT statement does not specify a value for the column.

**check.** Any CHECK condition defined on the column.

**user\_type** If the column is defined on a user-defined data type, the data type is held here.

**format\_str** Currently unused.

**column\_type** The type of column.

**remote\_name** The name of the remote column.

**remote\_type** The type of the remote column. This value is defined by the remote server or interface.

## SYSDOMAIN system table

Column name	Column type	Column constraint	Table constraints
domain_id	SMALLINT	NOT NULL	Primary key
domain_name	CHAR(128)	NOT NULL	
type_id	SMALLINT	NOT NULL	
precision	SMALLINT		

Each of the predefined data types (sometimes called **domains**) is assigned a unique number. The **SYSDOMAIN** table is provided for informational purposes, to show the association between these numbers and the appropriate data types. This table is never changed.

**domain\_id** The unique number assigned to each data type. These numbers cannot be changed.

**domain\_name** A string containing the data type normally found in the CREATE TABLE command, such as **char** or **integer**.

**type\_id** The ODBC data type. This corresponds to "data\_type" in the Transact-SQL-compatibility DBO.SYSTYPES table.

**precision** The number of significant digits that can be stored using this data type. The column value is NULL for non-numeric data types.

## SYSEXTENT system table

Column name	Column type	Column constraint	Table constraint
file_id	SMALLINT	NOT NULL	Primary key, foreign key references SYS.SYSFILE
extent_id	SMALLINT	NOT NULL	Primary key
first_page	INTEGER	NOT NULL	
last_page	INTEGER	NOT NULL	
file_name	LONG VARCHAR	NOT NULL	

This table is not used by Adaptive Server Anywhere 6.0.



## SYSEXTERNLOGINS system table

Column name	Column type	Column constraint	Table constraints
user_id	SMALLINT	NOT NULL	Primary key
srvid	INTEGER	NOT NULL	Primary key
remote_login	VARCHAR(128)		
remote_password	VARBINARY(128)		

This table is not used by Adaptive Server Anywhere 6.0.

## SYSFILE system table

Column name	Column type	Column constraint	Table constraint
file_id	SMALLINT	NOT NULL	Primary key
file_name	LONG VARCHAR	NOT NULL	
dbspace_name	CHAR(128)	NOT NULL	

Every database consists of one or more operating system files. Each file is recorded in **SYSFILE**.

**file\_id** Each file in a database is assigned a unique number. This file identifier is the primary key for **SYSFILE**. All system tables are stored in **file\_id** 0.

**file\_name** The database name is stored when a database is created. This name is for informational purposes only.

**dbspace\_name** Every file has a dbspace name that is unique. It is used in the CREATE TABLE command.

## SYSFKCOL system table

Column name	Column type	Column constraint	Table constraints
foreign_table_id	SMALLINT	NOT NULL	Primary key, foreign key references SYS.SYSCOLU MN.table_id
foreign_key_id	SMALLINT	NOT NULL	Primary key, foreign key references SYS.SYSFOREI GNKEY. foreign_key_id
foreign_column_id	SMALLINT	NOT NULL	Primary key, Foreign key references SYS.SYSCOLU MN.column_id
primary_column_id	SMALLINT	NOT NULL	

Each row of **SYSFKCOL** describes the association between a **foreign column** in the foreign table of a relationship and the **primary column** in the primary table.

**foreign\_table\_id** The table number of the foreign table.

**foreign\_key\_id** The key number of the FOREIGN KEY for the foreign table. Together, **foreign\_table\_id** and **foreign\_key\_id** uniquely identify one row in **SYSFOREIGNKEY**. The table number for the primary table can be obtained from that row (using the **SYSFOREIGNKEY** table).

**foreign\_column\_id** This column number, and the **foreign\_table\_id** identify the foreign column description in **SYSCOLUMN**.

**primary\_column\_id** This column number and the **primary\_table\_id** obtained from **SYSFOREIGNKEY** identify the primary column description in **SYSCOLUMN**.

## SYSFOREIGNKEY system table

Column name	Column type	Column constraint	Table constraints
foreign_table_id	SMALLINT	NOT NULL	Primary key, foreign key references SYS.SYSTABL E. table_id. Unique index.
foreign_key_id	SMALLINT	NOT NULL	Primary key
primary_table_id	SMALLINT	NOT NULL	foreign key references SYS.SYSTABL E. table_id.
root	INTEGER	NOT NULL	
check_on_commit	CHAR(1)	NOT NULL	
nulls	CHAR(1)	NOT NULL	
role	CHAR(128)	NOT NULL	Unique index
remarks	LONG VARCHAR		

A foreign key is a **relationship** between two tables—the **foreign table** and the **primary table**. Every foreign key is defined by one row in **SYSFOREIGNKEY** and one or more rows in **SYSFKCOL**. **SYSFOREIGNKEY** contains general information about the foreign key while **SYSFKCOL** identifies the columns in the foreign key and associates each column in the foreign key with a column in the primary key of the primary table.

**foreign\_table\_id** The table number of the foreign table.

**foreign\_key\_id** Each foreign key has a **foreign key number** that is unique with respect to:

- ◆ The key number of all other foreign keys for the foreign table
- ◆ The key number of all foreign keys for the primary table
- ◆ The index number of all indexes for the foreign table

**primary\_table\_id** The table number of the primary table.

**root** Foreign keys are stored in the database as B-trees. The **root** identifies the location of the root of the B-tree in the database file.

**check\_on\_commit (Y/N)** Indicates whether INSERT and UPDATE commands should wait until the next COMMIT command to check if foreign keys are valid. A foreign key is valid if, for each row in the foreign table, the values in the columns of the foreign key either contain the NULL value or match the primary key values in some row of the primary table.

**nulls (Y/N)** Indicates whether the columns in the foreign key are allowed to contain the NULL value. Note that this setting is independent of the **nulls** setting in the columns contained in the foreign key.

**role** The name of the relationship between the foreign table and the primary table. Unless otherwise specified, the **role** name will be the same as the name of the primary table. The foreign table cannot have two foreign keys with the same role name.

**remarks** A comment string.

## SYSGROUP system table

Column name	Column type	Column constraint	Table constraints
group_id	SMALLINT	NOT NULL	Primary key, foreign key references SYS.SYSUSER PERM. user_id
group_member	SMALLINT	NOT NULL	Primary key, foreign key references SYS.SYSUSER PERM. user_id

There is one row in **SYSGROUP** for every member of every group. This table describes a many-to-many relationship between groups and members. A group may have many members, and a user may be a member of many groups.

**group\_id** The user number of group.

**group\_member** The user number of a member.

## SYSINDEX system table

Column name	Column type	Column constraint	Table constraints
table_id	SMALLINT	NOT NULL	Primary key, Unique index
index_id	SMALLINT	NOT NULL	Primary key
root	INTEGER	NOT NULL	
file_id	SMALLINT	NOT NULL	
"unique"	CHAR(1)	NOT NULL	
creator	SMALLINT	NOT NULL	Foreign key references SYS.SYSUSER PERM. user_id
index_name	CHAR(128)	NOT NULL	Unique index
remarks	LONG, VARCHAR		

Each index in the database is described by one row in **SYSINDEX**. Each column in the index is described by one row in **SYSIXCOL**.

**table\_id** Uniquely identifies the table to which this index applies.

**index\_id** Each index for one particular table is assigned a unique index number.

**root** Indexes are stored in the database as B-trees. The **root** identifies the location of the root of the B-tree in the database file.

**file\_id** The index is completely contained in the file with this **file\_id** (see **SYSFILE**).

**unique** Indicate whether the index is a unique index ("Y"), a non-unique index ("N"), or a unique constraint ("U"). A unique index prevents two rows in the indexed table from having the same values in the index columns.

**creator** The user number of the creator of the index. This user is always the same as the creator of the table identified by table\_id.

**index\_name** The name of the index. A user ID cannot have two indexes with the same name in tables that it owns..

**remarks** A comment string.

## SYSINFO system table

Column name	Column type	Column constraint	Table constraints
page_size	SMALLINT	NOT NULL	
encryption	CHAR(1)	NOT NULL	
blank_padding	CHAR(1)	NOT NULL	
case_sensitivity	CHAR(1)	NOT NULL	
default_collation	CHAR(10)		
database_version	SMALLINT	NOT NULL	
classes_version	CHAR(10)		

This table indicates the database characteristics, as defined when the database was created. It always contains only one row.

**page\_size** The page size specified, in bytes. The default value is 1024.

**encryption** (Y/N) Indicates whether the -e switch was used with DBINIT.

**blank\_padding** (Y/N) Indicates whether the database was created to use blank padding for string comparisons in the database ( -b switch was used with *dbinit*).

**case\_sensitivity** (Y/N) Indicates whether the database is created as case sensitive. Case sensitivity affects value comparisons, but not table and column name comparisons. For example, if a database is case sensitive, table names such as **SYSCATALOG** can be specified in either case, but in a case-sensitive database '**abc**' = '**ABC**' is not true.

**default\_collation** A string corresponding to the **collation\_label** in **SYSCOLLATE**, which also corresponds to the collation sequence specified with DBINIT. The default value corresponds to the multilingual collation sequence (code page 850), which was the default prior to Watcom SQL 3.2. The collation sequence is used for all string comparisons, including searches for character strings as well as column and table name comparison.

**database\_version** A small integer value indicating the database format. As newer versions become available, new features may require that the format of the database file change. The version number Adaptive Server Anywhere software to determine if this database was created with a newer version of the software and thus, cannot be understood by the software in use.



**classes\_version** A small string describing the current version of the **SYS.JAVA.CLASSES** library that is currently installed on your computer.

## SYSIXCOL system table

Column name	Column type	Column constraint	Table constraints
table_id	SMALLINT	NOT NULL	Primary key, foreign key references SYS.SYSCOLUMN
index_id	SMALLINT	NOT NULL	
sequence	SMALLINT	NOT NULL	Primary key
column_id	SMALLINT	NOT NULL	
"order"	CHAR(1)	NOT NULL	

Every index has one row in **SYSIXCOL** for each column in the index.

**table\_id** Identifies the table to which the index applies.

**index\_id** Identifies in which index this column is used. Together, **table\_id** and **index\_id** identify one index described in **SYSINDEX**.

**sequence** Each column in an index is assigned a unique number starting at 0. The order of these numbers determines the relative significance of the columns in the index. The most important column has **sequence** number 0.

**column\_id** The column number identifies which column is indexed. Together, **table\_id** and **column\_id** identify one column in **SYSCOLUMN**.

**order (A/D)** Indicate whether this column in the index is kept in ascending or descending order.

## SYSJAR system table

Column name	Column type	Column constraint	Table constraints
jar_id	INTEGER	NOT NULL	Primary key
creator	SMALLINT	NOT NULL	
jar_name	LONG VARCHAR	NOT NULL	
jar_file	LONG VARCHAR		
create_time	TIMESTAMP	NOT NULL	
update_time	TIMESTAMP	NOT NULL	
remarks	LONG VARCHAR		

**jar\_id** A field containing the id of the jar file. This field also references the **SYSJAR** system table.

**creator** The is of the creator of the jar file.

**jar\_name** The name of the jar file.

**jar\_file** The file name of the jar file.

**create\_time** The time the jar file was created.

**update\_time** The time the jar file was last updated.

**remarks** A comment field.

## SYSJARCOMPONENT system table

Column name	Column type	Column constraint	Table constraints
component_id	INTEGER	NOT NULL	Primary key
jar_id	INTEGER		Foreign key references SYS.SYSJAR
component_name	LONG VARCHAR		
Component_type	CHAR(1)		
Create_time	TIMESTAMP	NOT NULL	
Contents	LONG BINARY		
Remarks	LONG VARCHAR		

**component\_id** The primary key containing the id of the component.

**jar\_id** A field containing the ID number of the jar. This field also references the **SYSJAR** system table.

**component\_name** The name of the component.

**component\_type** The type of the component.

**create\_time** A field containing the creation time of the component.

**contents** The byte code of the jar file.

**remarks** A comment field.

## SYSJAVACLASS system table

Column name	Column type	Column constraint	Table constraints
class_id	INTEGER	NOT NULL	Primary key
replaced_by	SMALLINT		Foreign key references SYS.SYSJAVA CLASSES. class_id
creator	SMALLINT	NOT NULL	Foreign key references SYS.SYSUSER PERM. user_id
jar_id	INTEGER		
type_id	SMALLINT		Foreign key references SYS.SYSUSER TYPE
class_name	LONG VARCHAR	NOT NULL	
public	CHAR(1)	NOT NULL	
component_id	INTEGER		Foreign key references SYS.SYSJARC OMONENT
create_time	TIMESTAMP	NOT NULL	
update_time	TIMESTAMP	NOT NULL	
class_descriptor	LONG BINARY		
remarks	LONG VARCHAR		

The **SYSJAVACLASS** system table contains all information related to Java classes.

**class\_id** This field contains the id of the java class. Also the primary key for the table.

**replaced\_by** A field that references the primary key field, class\_id.

**creator** This field contains the user\_id of the creator of the class. This field references the user\_id field in the **SYS.SYSUSERPERM** system table to obtain the name of the user.

**jar\_id** This field contains the id of the jar file from which the class came.

**type\_id** This field contains the id of the user type. This field references the SYS.SYSUSERTYPE system table to obtain the id of the user.

**class\_name** This field contains the name of the Java class.

**public** This field determines whether or not the class is public or private.

**component\_id** This field, which references the SYS.SYSJARCOMPONENT system table contains the id of the component.

**create\_time** Contains the creation time of the component.

**update\_time** Contains the last update time of the component.

**class\_descriptor** The byte code of the jar file.

**remarks** Contains a comment string.

## SYSLOGIN system table

Column name	Column type	Column constraint	Table constraints
integrated_login_id	CHAR(128)	NOT NULL	Primary key, Foreign key references SYS.SYSLOGIN. user_id.
login_uid	SMALLINT	NOT NULL	
remarks	LONG, VARCHAR		

This table contains all the User Profile names that can be used to connect to the database using an integrated logon. As a security measure, only users with DBA authority can view the contents of this table.

**integrated\_login\_id** A string value containing the User Profile name that is used to map to a user ID in the database. When a user successfully logs on using this User Profile name, and the database is enabled to accept integrated logons, the user can connect to the database without providing a user ID or password.

**login\_uid** A foreign key to the system table **SYSUSERPERM**.

**remarks** A comment string

## SYSOPTION system table

Column name	Column type	Column constraint	Table constraints
user_id	SMALLINT	NOT NULL	Primary key, foreign key references SYS.SYSUSER PERM
"option"	CHAR(128)	NOT NULL	Primary key
"setting"	LONG VARCHAR	NOT NULL	

Options settings are stored in the **SYSOPTION** table by the SET command. Each user can have their own setting for each option. In addition, settings for the **PUBLIC** user ID define the default settings to be used for user IDs that do not have their own setting.

**user\_id** The user number to whom this option setting applies.

**option** The name of the option.

**setting** The current setting for the named option.



## SYSPROCEDURE system table

Column name	Column type	Column constraint	Table constraints
proc_id	SMALLINT	NOT NULL	Primary key  Foreign key references SYS.SYSUSER PERM. user_id
creator	SMALLINT	NOT NULL	
proc_name	CHAR(128)	NOT NULL	
proc_defn	LONG VARCHAR		
remarks	LONG VARCHAR		
replicate	CHAR(1)	NOT NULL	

Each procedure in the database is described by one row in **SYSPROCEDURE**.

**proc\_id** Each procedure is assigned a unique number (the **procedure number**), which is the primary key for **SYSPROCEDURE**.

**creator** This user number identifies the owner of the procedure. The name of the user can be found by looking in **SYSUSERPERM**.

**proc\_name** The name of the procedure. One creator cannot have two procedures with the same name.

**proc\_defn** The command that was used to create the procedure.

**remarks** A comment string.

**replicate** (Y/N) Indicates whether the procedure is a primary data source in a Replication Server installation.

## SYSPROCPARM system table

Column name	Column type	Column constraint	Table constraint
proc_id	SMALLINT	NOT NULL	Primary key, foreign key references SYS.SYSPROCEDURE
parm_id	SMALLINT	NOT NULL	Primary key
parm_type	SMALLINT	NOT NULL	
parm_mode_in	CHAR(1)	NOT NULL	
parm_mode_out	CHAR(1)	NOT NULL	
domain_id	SMALLINT	NOT NULL	Foreign key references SYS.SYSDOMAIN
width	SMALLINT	NOT NULL	
scale	SMALLINT	NOT NULL	
parm_name	CHAR(128)	NOT NULL	
remarks	LONG VARCHAR		
"default"	LONG VARCHAR		
user_type	INTEGER		

Each parameter to a procedure in the database is described by one row in **SYSPROCEDURE**.

**proc\_id** Uniquely identifies the procedure to which this parameter belongs.

**parm\_id** Each procedure starts numbering parameters at 1. The order of parameter numbers corresponds to the order in which they were defined.

**parm\_type** The type of parameter will be one of the following:

- ◆ Normal parameter (variable)
- ◆ Result variable - used with a procedure that return result sets
- ◆ SQLSTATE error value
- ◆ SQLCODE error value

**parm\_mode\_in (Y/N)** Indicates whether this parameter supplies a value to the procedure (**IN** or **INOUT** parameters).

**parm\_mode\_out (Y/N)** Indicates whether this parameter returns a value from the procedure (**OUT** or **INOUT** parameters).

**domain\_id** Identifies the data type for the parameter, by the data type number listed in the **SYSDOMAIN** table.

**width** Contains the length of a string parameter, the precision of a numeric parameter, or the number of bytes of storage for any other data types.

**scale** The number of digits after the decimal point for numeric data type parameters, and zero for all other data type.

**parm\_name** The name of the procedure parameter.

**remarks** A comment string.

**default** The default value for the parameter, held as a string.

**user\_type** The user type of the parameter.

## SYSPROCPERM system table

Column name	Column type	Column constraint	Table constraints
proc_id	SMALLINT	NOT NULL	Primary key
grantee	SMALLINT	NOT NULL	Primary key, foreign key references SYS.SYSUSER PERM. user_id

Only users who have been granted permission can call a procedure. Each row of the **SYSPROCPERM** table corresponds to one user granted permission to call one procedure.

**proc\_id** The procedure number uniquely identifies the procedure for which permission has been granted.

**grantee** The user number of the user ID receiving the permission.

## SYSPUBLICATION system table

Column name	Column type	Column constraint	Table constraints
publication_id	SMALLINT	NOT NULL	Primary key
creator	SMALLINT	NOT NULL	Foreign key references SYS.SYSUSER PERM. user_id
publication_name	CHAR(128)	NOT NULL	Unique index
remarks	LONG VARCHAR		

Each row describes a SQL Remote publication.

**publication\_id** A unique identifying number for the publication.

**creator** The owner of the publication.

**publication\_name** The name of the publication, which must be a valid identifier.

**remarks** Descriptive comments.

## SYSREMOTETYPE system table

Column name	Column type	Column constraint	Table constraints
type_id	SMALLINT	NOT NULL	Primary key
type_name	CHAR(128)	NOT NULL	
publisher_address	LONG VARCHAR	NOT NULL	
remarks	LONG VARCHAR		

The **SYSREMOTETYPE** system table contains information about SQL Remote.

**type\_id** Identifies which of the of the message systems supported by SQL Remote is to be used to send messages to this user.

**type\_name** The name of the message system supported by SQL Remote.

**publisher\_address** The address of the remote database publisher.

**remarks** Descriptive comments.

## SYSREMOTEUSER system table

Column name	Column type	Column constraint	Table constraints
user_id	SMALLINT	NOT NULL	Primary key, foreign key references SYS.SYSUSER PERM
consolidate	CHAR(1)	NOT NULL	
type_id	SMALLINT	NOT NULL	Foreign key references SYS.SYSREMO TETYPE
address	LONG, VARCHAR	NOT NULL	
frequency	CHAR(1)	NOT NULL	
send_time	TIME		
log_send	NUMERIC(20,0)	NOT NULL	
time_sent	TIMESTAMP		
log_sent	NUMERIC(20,0)	NOT NULL	
confirm_sent	NUMERIC(20,0)	NOT NULL	
send_count	INTEGER	NOT NULL	
resend_count	INTEGER	NOT NULL	
time_received	TIMESTAMP		
log_received	NUMERIC(20,0)	NOT NULL	
confirm_received	NUMERIC(20,0)		
receive_count	INTEGER	NOT NULL	
rereceive_count	INTEGER	NOT NULL	

Each row describes a userid with REMOTE permissions (a subscriber), together with the status of SQL Remote messages that were sent to and from that user.

**user\_id** The user number of the user with REMOTE permissions.

**consolidate** (Y/N) Indicates whether the user was granted CONSOLIDATE permissions (Y) or REMOTE permissions (N).

**type\_id** Identifies which of the of the message systems supported by SQL Remote is used to send messages to this user.

**address** The address to which SQL Remote messages are to be sent. The address must be appropriate for the **address\_type**.

**frequency** How frequently SQL Remote messages are sent.

**send\_time** The next time messages are to be sent to this user.

**log\_send** Messages are sent only to subscribers for whom **log\_send** is greater than **log\_sent**.

**time\_sent** The time the most recent message was sent to this subscriber.

**log\_sent** The log offset for the most recently sent operation.

**confirm\_sent** The log offset for the most recently confirmed operation from this subscriber.

**send\_count** How many SQL Remote messages have been sent.

**resend\_count** Counter to ensure that messages are applied only once at the subscriber database.

**time\_received** The time when the most recent message was received from this subscriber.

**log\_received** The log offset in the subscriber's database for the operation that was most recently received at the current database.

**confirm\_received** The log offset in the subscriber's database for the most recent operation for which a confirmation message has been sent.

**receive\_count** How many messages have been received.

**rereceive\_count** Counter to ensure that messages are applied only once at the current database.



## SYSSERVERS system table

Column name	Column type	Column constraint	Table Constraints
srvid	INTEGER	NOT NULL	Primary key
srvname	VARCHAR(128)	NOT NULL	
srvclass	LONG VARCHAR	NOT NULL	
srvinfo	LONG VARCHAR		
srvreadonly	CHAR(1)	NOT NULL	

The **SYSSERVERS** system table is not used by Adaptive Server Anywhere 6.0.

## SYSSQLSERVERTYPE system table

Column name	Column type	Column constraint	Table constraints
ss_user_type	SMALLINT	NOT NULL	Primary key
ss_domain_id	SMALLINT	NOT NULL	
ss_type_name	VARCHAR (30)	NOT NULL	
primary_sa_domain_id	SMALLINT	NOT NULL	
primary_sa_user_type	SMALLINT	NULL	

This table contains information relating to compatibility with Adaptive Server Enterprise.

**ss\_user\_type** A smallint field describing the Adaptive Server Enterprise user type

**ss\_domain\_id** A smallint field describing the Adaptive Server Enterprise domain id.

**ss\_type\_name** Contains the Adaptive Server Enterprise type name.

**primary\_sa\_domain\_id** A smallint field containing the Adaptive Server Anywhere primary domain id.

**primary\_sa\_user\_type** A smallint field containing the Adaptive Server Anywhere primary user type.

## SYSSUBSCRIPTION system table

Column name	Column type	Column constraint	Table constraints
publication_id	SMALLINT	NOT NULL	Primary key, foreign key references SYS.SYSSUBS CRPTION
user_id	SMALLINT	NOT NULL	
subscribe_by	CHAR(128)	NOT NULL	Primary key
created	NUMERIC(20,0)	NOT NULL	
started	NUMERIC(20,0)		

Each row describes a subscription from one user ID (which must have REMOTE permissions) to one publication.

**publication\_id** The identifier for the publication to which the user ID is subscribed.

**user\_id** The user number that is subscribed to the publication.

**subscribe\_by** The value of the SUBSCRIBE BY expression, if any, for the subscription.

**created** The offset in the transaction log at which the subscription was created.

**started** The offset in the transaction log at which the subscription was started.

## SYSTABLE system table

Column name	Column type	Column constraint	Table constraints
table_id	SMALLINT	NOT NULL	Primary key
file_id	SMALLINT	NOT NULL	Foreign key references SYS.SYSFILE
count	INTEGER	NOT NULL	
first_page	INTEGER	NOT NULL	
last_page	INTEGER	NOT NULL	
primary_root	INTEGER	NOT NULL	
creator	SMALLINT	NOT NULL	Foreign key references SYS.SYSUSERPERM . user_id
table_name	CHAR(128)	NOT NULL	
table_type	CHAR(10)	NOT NULL	
view_def	LONG VARCHAR		
remarks	LONG VARCHAR		
replicate	CHAR(1)	NOT NULL	
"existing_obj"	CHAR(1)		
remote_location	LONG VARCHAR		
remote_objtype	CHAR(1)		
srv_id	INTEGER		Foreign key references SYS.SYSSERVERS

Each row of **SYSTABLE** describes one table or view in the database.

**table\_id** Each table or view is assigned a unique number (the table number) which is the primary key for **SYSTABLE**.

**file\_id** Indicates which database file contains the table. The **file\_id** is a FOREIGN KEY for **SYSFILE**.

**count** The number of rows in the table is updated during each successful CHECKPOINT. This number is used by Adaptive Server Anywhere when optimizing database access. The **count** is always 0 for a view.

**first\_page** Each database is divided into a number of fixed-size pages. This value identifies the first page that contains information for this table, and is used internally to find the start of this table. The **first\_page** is always 0 for a view.

**last\_page** The last page that contains information for this table. The **last\_page** is always 0 for a view.

**primary\_root** Primary keys are stored in the database as B-trees. The **primary\_root** locates the root of the B-tree for the primary key for the table. It will be 0 for a view and, for a table with no primary key.

**creator** The user number of the owner of the table or view. The name of the user can be found by looking in **SYSUSERPERM**.

**table\_name** The name of the table or view. One creator cannot have two tables or views with the same name.

**table\_type** This column is **BASE** for base tables, **VIEW** for views, and be **GBL TEMP** for global temporary tables. No entry is created for local temporary tables.

**view\_def** For a view, this column contains the CREATE VIEW command that was used to create the view. For a table, this column contains any CHECK constraints for the table.

**remarks** A comment string.

**replicate** (Y/N) Indicates whether the table is a primary data source in a Replication Server installation.

**existing\_obj** (Y/N) Indicates whether the table previously existed or not.

**remote\_location** Indicates the storage location of the remote object.

**remote\_objtype** Indicates the type of remote object: 'T' if table; 'V' if view; 'R' if rpc; 'B' if JavaBean.

**srvid** The unique ID for the server.

## SYSTABLEPERM system table

Column name	Column type	Column constraint	Table constraints
stable_id	SMALLINT	NOT NULL	Primary key, foreign key references SYS.SYSTABLE table_id
grantee	SMALLINT	NOT NULL	Primary key, foreign key references SYS.SYSUSERP ERM. user_id
grantor	SMALLINT	NOT NULL	Primary key, foreign key references SYS.SYSUSERP ERM. user_id
ttable_id	SMALLINT	NOT NULL	Foreign key references SYS.SYSTABLE table_id
selectauth	CHAR(1)	NOT NULL	
insertauth	CHAR(1)	NOT NULL	
deleteauth	CHAR(1)	NOT NULL	
updateauth	CHAR(1)	NOT NULL	
updatecols	CHAR(1)	NOT NULL	
alterauth	CHAR(1)	NOT NULL	
referenceauth	CHAR(1)	NOT NULL	

Permissions given by the GRANT command are stored in **SYSTABLEPERM**. Each row in this table corresponds to one table, one user ID granting the permission (**grantor**) and one user ID granted the permission (**grantee**).

There are several types of permission that can be granted. Each permission can have one of the following three values.

- ◆ **N** No, the grantee has not been granted this permission by the grantor.
- ◆ **Y** Yes, the grantee has been given this permission by the grantor.

- ◆ **G** The grantee has been given this permission and can grant the same permission to another user (with grant options).

#### Permissions

The grantee might have been given permission for the same table by another grantor. If so, this information would be recorded in a different row of **SYSTABLEPERM**.

**stable\_id** The table number of the table or view to which the permissions apply.

**grantor** The user number of the user ID granting the permission.

**grantee** The user number of the user ID receiving the permission.

**ttable\_id** In the current version of Adaptive Server Anywhere, this table number is always the same as **stable\_id**.

**selectauth (Y/N/G)** Indicates whether SELECT permission has been granted.

**insertauth (Y/N/G)** Indicates whether INSERT permission has been granted.

**deleteauth (Y/N/G)** Indicates whether DELETE permission has been granted.

**updateauth (Y/N/G)** Indicates whether UPDATE permission has been granted for all columns in the table. (Only UPDATE permission can be given on individual columns. All other permissions are for all columns in a table.)

**updatecols (Y/N)** Indicates whether UPDATE permission has only been granted for some of the columns in the table. If **updatecols** has the value Y, there will be one or more rows in **SYSCOLPERM** granting update permission for the columns in this table.

**alterauth (Y/N/G)** Indicates whether ALTER permission has been granted.

**referenceauth (Y/N/G)** Indicates whether REFERENCE permission has been granted.

## SYSTRIGGER system table

Column name	Column type	Column constraint	Table constraints
trigger_id	SMALLINT	NOT NULL	Primary key
table_id	SMALLINT	NOT NULL	Foreign key references SYS.SYSTABLE. table_id
event	CHAR(1)	NOT NULL	
trigger_time	CHAR(1)	NOT NULL	
trigger_order	SMALLINT		
foreign_table_id	SMALLINT		Foreign key references SYS.SYSFOREIGN KEY
foreign_key_id	SMALLINT		Foreign key references SYS.SYSFOREIGN KEY
referential_action	CHAR(1)		
trigger_name	CHAR(128)		
trigger_defn	LONG VARCHAR	NOT NULL	
remarks	LONG VARCHAR		

Each trigger in the database is described by one row in **SYSTRIGGER**. The table also contains triggers that are automatically created by the database for foreign key definitions which have a referential triggered action (such as ON DELETE CASCADE).

**trigger\_id** Each trigger is assigned a unique number (the **trigger number**), which is the primary key for **SYSTRIGGER**.

**table\_id** The table number uniquely identifies the table to which this trigger belongs.

**event** The event or events that cause the trigger to fire. This single-character value corresponds to the trigger event that was specified when the trigger was created.



**trigger\_time** The time at which the trigger will fire. This single-character value corresponds to the trigger time that was specified when the trigger was created.

- ◆ **A** AFTER
- ◆ **B** BEFORE

**trigger\_order** The order in which the trigger will fire. This determines the order that triggers are fired when there are triggers of the same type (insert, update, or delete) that fire at the same time (before or after).

**foreign\_table\_id** The table number of the table containing a foreign key definition which has a referential triggered action (such as ON DELETE CASCADE).

**foreign\_key\_id** The foreign key number of the foreign key for the table referenced by **foreign\_table\_id**.

**referential\_action** The action defined by a foreign key. This single-character value corresponds to the action that was specified when the foreign key was created.

- ◆ **C** CASCADE
- ◆ **D** SET DEFAULT
- ◆ **N** SET NULL
- ◆ **R** RESTRICT

**trigger\_name** The name of the trigger. One table cannot have two triggers with the same name.

**trigger\_defn** The command that was used to create the trigger.

**remarks** A comment string.

## SYSTYPEMAP system table

Column name	Column type	Column constraint	Table constraints
ss_user_type	SMALLINT	NOT NULL	Foreign key references SYS.SYSSQLSERVER.ERVERTYPE
sa_domain_id	SMALLINT	NOT NULL	
sa_user_type	SMALLINT	NULL	
nullable	CHAR(1)	NULL	

The SYSTYPEMAP system table contains the compatibility mapping values for the SYSSQLSERVER.ERVERTYPE system table.

**ss\_user\_type** Contains the Adaptive Server Enterprise user type.

**sa\_domain\_id** Contains the Adaptive Server Anywhere 6.0 domain\_id.

**sa\_user\_type** Contains the Adaptive Server Anywhere 6.0 user type.

**nullable** This field describes whether or not the type can or cannot be null.

## SYSUSERMESSAGES system table

Column name	Column type	Column constraint	Table constraints
error	INTEGER	NOT NULL	Unique index
uid	SMALLINT	NOT NULL	
description	VARCHAR(255)	NOT NULL	
langid	SMALLINT	NOT NULL	Unique index

Each row holds a user-defined message for an error condition.

**error** A unique identifying number for the error condition.

**uid** The user number that defined the message.

**description** The message corresponding to the error condition.

**langid** Reserved.

## SYSUSERPERM system table

Column name	Column type	Column constraint	Table constraints
user_id	SMALLINT	NOT NULL	Primary key
user_name	CHAR(128)	NOT NULL	
password	BINARY(36)		
resourceauth	CHAR(1)	NOT NULL	
dbauth	CHAR(1)	NOT NULL	
scheduleauth	CHAR(1)	NOT NULL	
publishauth	CHAR(1)	NOT NULL	
remotedbauth	CHAR(1)	NOT NULL	
user_group	CHAR(1)	NOT NULL	
remarks	LONG VARCHAR		

### DBA permissions required

SYSUSERPERM contains passwords, so DBA permissions are required to SELECT from it.

Each row of **SYSUSERPERM** describes one user ID.

**user\_id** Each new user ID is assigned a unique number (the **user number**), which is the primary key for **SYSUSERPERM**.

**user\_name** A string containing a unique name for the user ID.

**password** The password for the user ID. The password contains the NULL value for the special user IDs **SYS** and **PUBLIC**. This prevents anyone from connecting to these user IDs.

**resourceauth (Y/N)** Indicates whether the user has RESOURCE authority. Resource authority is required to create tables.

**dbauth (Y/N)** Indicates whether the user has DBA (database administrator) authority. DBA authority is very powerful, and should be restricted to as few user IDs as possible for security purposes.

**scheduleauth (Y/N)** Indicates whether the user has SCHEDULE authority. This is currently not used.

**publishauth (Y/N)** Indicates whether the user has the SQL Remote publisher authority.

**remotedbaauth (Y/N)** Indicates whether the user has the SQL Remote remote DBA authority.

**user\_group (Y/N)** Indicates whether the user is a group.

**remarks** A comment string.

When a database is initialized, the following user IDs are created:

- ◆ **SYS** The creator of all the system tables.
- ◆ **PUBLIC** A special user ID used to record PUBLIC permissions.
- ◆ **DBA** The database administrator user ID is the only usable user ID in an initialized system. The initial password is SQL.

There is no way to connect to the **SYS** or **PUBLIC** user IDs.

## SYSUSERTYPE system table

Column name	Column type	Column constraint	Table constraints
type_id	SMALLINT	NOT NULL	Primary key
creator	SMALLINT	NOT NULL	Foreign key references SYS.SYSUSER PERM. user_id
domain_id	SMALLINT	NOT NULL	Foreign key references SYS.SYSDOM AIN
nulls	CHAR(1)	NOT NULL	
width	SMALLINT	NOT NULL	
scale	SMALLINT	NOT NULL	
type_name	CHAR(128)	NOT NULL	
"default"	LONG VARCHAR	NULL	
"check"	LONG VARCHAR	NULL	
format_str	CHAR(128)		
super_type_id	SMALLINT	NULL	Foreign key references SYS.SYSUSER TYPE. type_id.

Each row holds a description of a user-defined data type.

**type\_id** A unique identifying number for the user-defined data type.

**creator** The user number of the owner of the data type.

**domain\_id** The data type on which this user defined data type is based, indicated by a data type number listed in the **SYSDOMAIN** table.

**nulls** (Y/N) Indicates whether the user-defined data type allows nulls.

**width** The length of a string column, the precision of a numeric column, or the number of bytes of storage for any other data type.

**scale** The number of digits after the decimal point for numeric data type columns, and zero for all other data types.

**type\_name** The name for the data type, which must be a valid identifier.

**default** The default value for the data type.

**check** The CHECK condition for the data type.

**format\_str** Currently unused.

