



# **Quick Reference Guide**

**Adaptive Server® Enterprise**

**12.5.1**

DOCUMENT ID: DC70202-01-1251-01

LAST REVISED: August 2003

Copyright © 1989-2003 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

To order additional documents, U.S. and Canadian customers should call Customer Fulfillment at (800) 685-8225, fax (617) 229-9845.

Customers in other countries with a U.S. license agreement may contact Customer Fulfillment via the above fax number. All other international customers should contact their Sybase subsidiary or local distributor. Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase, the Sybase logo, AccelaTrade, ADA Workbench, Adaptable Windowing Environment, Adaptive Component Architecture, Adaptive Server, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Server Enterprise Monitor, Adaptive Server Enterprise Replication, Adaptive Server Everywhere, Adaptive Server IQ, Adaptive Warehouse, Anywhere Studio, Application Manager, AppModeler, APT Workbench, APT-Build, APT-Edit, APT-Execute, APT-FORMS, APT-Translator, APT-Library, AvantGo, AvantGo Application Alerts, AvantGo Mobile Delivery, AvantGo Mobile Document Viewer, AvantGo Mobile Inspection, AvantGo Mobile Marketing Channel, AvantGo Mobile Pharma, AvantGo Mobile Sales, AvantGo Pylon, AvantGo Pylon Application Server, AvantGo Pylon Conduit, AvantGo Pylon PIM Server, AvantGo Pylon Pro, Backup Server, BizTracker, ClearConnect, Client-Library, Client Services, Convoy/DM, Copernicus, Data Pipeline, Data Workbench, DataArchitect, Database Analyzer, DataExpress, DataServer, DataWindow, DB-Library, dbQueue, Developers Workbench, Direct Connect Anywhere, DirectConnect, Distribution Director, e-ADK, E-Anywhere, e-Biz Integrator, E-Whatever, EC Gateway, ECMAP, ECRTP, eFulfillment Accelerator, Embedded SQL, EMS, Enterprise Application Studio, Enterprise Client/Server, Enterprise Connect, Enterprise Data Studio, Enterprise Manager, Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, eProcurement Accelerator, EWA, Financial Fusion, Financial Fusion Server, Gateway Manager, GlobalFIX, ImpactNow, Industry Warehouse Studio, InfoMaker, Information Anywhere, Information Everywhere, InformationConnect, InternetBuilder, iScript, Jaguar CTS, jConnect for JDBC, Mail Anywhere Studio, MainframeConnect, Maintenance Express, Manage Anywhere Studio, M-Business Channel, M-Business Network, M-Business Server, MDI Access Server, MDI Database Gateway, media.splash, MetaWorks, My AvantGo, My AvantGo Media Channel, My AvantGo Mobile Marketing, MySupport, Net-Gateway, Net-Library, New Era of Networks, ObjectConnect, ObjectCycle, OmniConnect, OmniSQL Access Module, OmniSQL Toolkit, Open Biz, Open Client, Open ClientConnect, Open Client/Server, Open Client/Server Interfaces, Open Gateway, Open Server, Open ServerConnect, Open Solutions, Optima++, PB-Gen, PC APT Execute, PC Net Library, PocketBuilder, Pocket PowerBuilder, Power++, power.stop, PowerAMC, PowerBuilder, PowerBuilder Foundation Class Library, PowerDesigner, PowerDimensions, PowerDynamo, PowerJ, PowerScript, PowerSite, PowerSocket, Powersoft, PowerStage, PowerStudio, PowerTips, Powersoft Portfolio, Powersoft Professional, PowerWare Desktop, PowerWare Enterprise, ProcessAnalyst, Rapport, Report Workbench, Report-Execute, Replication Agent, Replication Driver, Replication Server, Replication Server Manager, Replication Toolkit, Resource Manager, RW-DisplayLib, S-Designer, SDF, Secure SQL Server, Secure SQL Toolset, Security Guardian, SKILLS, smart.partners, smart.parts, smart.script, SQL Advantage, SQL Anywhere, SQL Anywhere Studio, SQL Code Checker, SQL Debug, SQL Edit, SQL Edit/TPU, SQL Everywhere, SQL Modeler, SQL Remote, SQL Server, SQL Server Manager, SQL SMART, SQL Toolset, SQL Server/CFT, SQL Server/DBM, SQL Server SNMP SubAgent, SQL Station, SQLJ, STEP, SupportNow, S.W.I.F.T. Message Format Libraries, Sybase Central, Sybase Client/Server Interfaces, Sybase Financial Server, Sybase Gateways, Sybase MPP, Sybase SQL Desktop, Sybase SQL Lifecycle, Sybase SQL Workgroup, Sybase User Workbench, SybaseWare, Syber Financial, SyberAssist, SyBooks, System 10, System 11, System XI (logo), SystemTools, Tabular Data Stream, TradeForce, Transact-SQL, Translation Toolkit, UltraLite.NET, UNIBOM, Unilib, Uninull, Unisep, Unistring, URK Runtime Kit for UniCode, Viewer, Visual Components, VisualSpeller, VisualWriter, VQL, WarehouseArchitect, Warehouse Control Center, Warehouse Studio, Warehouse WORKS, Watcom, Watcom SQL, Watcom SQL Server, Web Deployment Kit, Web.PB, Web.SQL, WebSights, WebViewer, WorkGroup SQL Server, XA-Library, XA-Server and XP Server are trademarks of Sybase, Inc. 03/03

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names used herein may be trademarks or registered trademarks of their respective companies.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

---

# Quick Reference

Topic	Page
Datatypes	1
Reserved words	3
Global variables	5
Functions	10
Commands	14
System procedures	33
Catalog stored procedures	42
System Extended Stored Procedures	43
dbcc stored procedures	44
System tables	45
dbcc tables	48
Utilities	49

## Datatypes

See *Reference Manual: Building Blocks* for more details.

Datatypes	Synonyms	Range	Bytes of storage
<i>Exact numeric datatypes</i>			
tinyint		0 to 255	1
smallint		$-2^{15}$ (-32,768) to $2^{15} - 1$ (32,767)	2
int	integer	$-2^{31}$ (-2,147,483,648) to $2^{31} - 1$ (2,147,483,647)	4
numeric (p, s)		$-10^{38}$ to $10^{38} - 1$	2 to 17
decimal (p, s)	dec	$-10^{38}$ to $10^{38} - 1$	2 to 17
<i>Approximate numeric datatypes</i>			
float (precision)		Machine dependent	4 or 8
double precision		Machine dependent	8

## Datatypes

---

Datatypes	Synonyms	Range	Bytes of storage
real		Machine dependent	4
<i>Money datatypes</i>			
smallmoney		-214,748.3648 to 214,748.3647	4
money		-922,337,203,685,477.5808 to 922,337,203,685,477.5807	8
<i>Date/time datatypes</i>			
smalldatetime		January 1, 1900 to June 6, 2079	4
datetime		January 1, 1753 to December 31, 9999	8
date		January 1 0001 to December 31, 9999	8
time		00:00:00:000 to 23:59:59:999	4
<i>Character datatypes</i>			
char(n)	character	Determined by your server's logical page size	<i>n</i>
varchar(n)	char[acter] varying	Determined by your server's logical page size	actual entry length
unicar	Unicode character	Determined by your server's logical page size	<i>n</i> *@@ <i>unicarsize</i> (@@ <i>unicarsize</i> equals 2)
univarchar	Unicode character varying	Determined by your server's logical page size	actual number of characters *@@ <i>unicarsize</i>
nchar(n)	national char[acter]	Determined by your server's logical page size	<i>n</i> * @@ <i>ncharsize</i>
nvarchar(n)	nchar varying, national char[acter] varying	Determined by your server's logical page size	<i>n</i>
<i>Binary datatypes</i>			
binary(n)		Determined by your server's logical page size	<i>n</i>
varbinary(n)		Determined by your server's logical page size	actual entry length
<i>Bit datatype</i>			
bit		0 or 1	1 (1 byte holds up to 8 bit columns)
<i>Text and image datatypes</i>			
text		2 <sup>31</sup> -1 (2,147,483,647) bytes or fewer	0 until initialized, then a multiple of the logical page size
image		2 <sup>31</sup> -1 (2,147,483,647) bytes or fewer	0 until initialized, then a multiple of the logical page size

---

## Reserved words

See *Reference Manual: Building Blocks* for details.

### Transact-SQL reserved words

	Words
<i>A</i>	add, all, alter, and, any, arith_overflow, as, asc, at, authorization, avg
<i>B</i>	begin, between, break, browse, bulk, by
<i>C</i>	cascade, case, char_convert, check, checkpoint, close, clustered, coalesce, commit, compute, confirm, connect, constraint, continue, controlrow, convert, count, create, current, cursor
<i>D</i>	database, dbcc, deallocate, declare, default, delete, desc, deterministic, disk distinct, double, drop, dummy, dump
<i>E</i>	else, end, endtran, errlvl, errordata, errexit, escape, except, exclusive, exec, execute, exists, exit, exp_row_size, external
<i>F</i>	fetch, fillfactor, for, foreign, from, func
<i>G</i>	goto, grant, group
<i>H</i>	having, holdlock
<i>I</i>	identity, identity_gap, identity_insert, identity_start, if, in, index, inout, insert, install, intersect, into, is, isolation
<i>J</i>	jar, join
<i>K</i>	key, kill
<i>L</i>	level, like, lineno, load, lock
<i>M</i>	max, max_rows_per_page, min, mirror, mirrorexit, modify
<i>N</i>	national, new, noholdlock, nonclustered, not, null, nullif, numeric_truncation
<i>O</i>	of, off, offsets, on, once, online, only, open, option, or, order, out, output, over
<i>P</i>	partition, perm, permanent, plan, precision, prepare, primary, print, privileges, proc, procedure, processexit, proxy_table, public
<i>Q</i>	quiesce
<i>R</i>	raiserror, read, readpast, readtext, reconfigure, references remove, reorg, replace, replication, reservepagegap, return, returns, revoke, role, rollback, rowcount, rows, rule
<i>S</i>	save, schema, select, set, setuser, shared, shutdown, some, statistics, stringsize, stripe, sum, syb_identity, syb_restree, syb_terminate
<i>T</i>	table, temp, temporary, textsize, to, tran, transaction, trigger, truncate, tsequal
<i>U</i>	union, unique, unpartition, update, use, user, user_option, using
<i>V</i>	values, varying, view
<i>W</i>	waitfor, when, where, while, with, work, writetext

## SQL92 reserved words

	Words
<i>A</i>	absolute, action, allocate, are, assertion
<i>B</i>	bit, bit_length, both
<i>C</i>	cascaded, case, cast, catalog, char, char_length, character, character_length, coalesce, collate, collation, column, connection, constraints, corresponding, cross, current_date, current_time, current_timestamp, current_user
<i>D</i>	date, day, dec, decimal, deferrable, deferred, describe, descriptor, diagnostics, disconnect, domain
<i>E</i>	end-exec, exception, extract
<i>F</i>	false, first, float, found, full
<i>G</i>	get, global, go
<i>H</i>	hour
<i>I</i>	immediate, indicator, initially, inner, input, insensitive, int, integer, interval
<i>J</i>	join
<i>L</i>	language, last, leading, left, local, lower
<i>M</i>	match, minute, module, month
<i>N</i>	names, natural, nchar, next, no, nullif, numeric
<i>O</i>	octet_length, outer, output, overlaps
<i>P</i>	pad, partial, position, preserve, prior
<i>R</i>	real, relative, restrict, right
<i>S</i>	scroll, second, section, session_user, size, smallint, space, sql, sqlcode, sqlerror, sqlstate, substring, system_user
<i>T</i>	then, time, timestamp, timezone_hour, timezone_minute, trailing, translate, translation, trim, true
<i>U</i>	unknown, upper, usage
<i>V</i>	value, varchar
<i>W</i>	when, whenever, write, year
<i>Z</i>	zone

## Potential SQL92 reserved words

	Words
<i>A</i>	after, alias, async
<i>B</i>	before, boolean, breadth
<i>C</i>	call, completion, cycle
<i>D</i>	data, depth, dictionary
<i>E</i>	each, elseif, equals

---

	<b>Words</b>
<i>G</i>	general
<i>I</i>	ignore
<i>L</i>	leave, less, limit, loop
<i>M</i>	modify
<i>N</i>	new, none
<i>O</i>	object, oid, old, operation, operators, others
<i>P</i>	parameters, pendant, preorder, private, protected
<i>R</i>	recursive, ref, referencing, resignal, return, returns, routine, row
<i>S</i>	savepoint, search, sensitive, sequence, signal, similar, sqlexception, structure
<i>T</i>	test, there, type
<i>U</i>	under
<i>V</i>	variable, virtual, visible
<i>W</i>	wait, without

## Global variables

See *Reference Manual: Building Blocks* for details.

<code>@ @bootcount</code>	Returns the number of times an Adaptive Server installation has been booted.
<code>@ @boottime</code>	Returns the date and time Adaptive Server was last booted.
<code>@ @bulkarraysize</code>	Returns the number of rows to be buffered in local server memory before being transferred using the bulk copy interface Used only with Component Integration Services for transferring rows to a remote server using select into.
<code>@ @bulkbatchsize</code>	Returns the number of rows transferred to a remote server via select into <i>proxy_table</i> using the bulk interface. Used only with Component Integration Services for transferring rows to a remote server using select into.
<code>@ @char_convert</code>	Returns 0 if character set conversion is not in effect. Returns 1 if character set conversion is in effect.
<code>@ @cis_rpc_handling</code>	Returns 0 if cis rpc handling is off. Returns 1 if cis rpc handling is on.
<code>@ @cis_version</code>	Returns the date and version of Component Integration Services.
<code>@ @client_csexpansion</code>	Returns the expansion factor used when converting from the server character set to the client character set.

@ @client_csid	Returns -1 if the client character set has never been initialized. Returns the client character set ID from syscharsets for the connection if the client character set has been initialized.
@ @client_csname	Returns NULL if client character set has never been initialized. Returns the name of the character set for the connection if the client character set has been initialized.
@ @cmpstate	Returns the current mode of Adaptive Server in a high availability environment.
@ @connections	Returns the number of user logins attempted.
@ @cpu_busy	Returns the number of seconds, in CPU time, that Adaptive Server's CPU was performing Adaptive Server work.
@ @curluid	Returns the current session's lock owner ID.
@ @datefirst	Set using set datefirst n where n is a value between 1 and 7. Returns the current value of @ @datefirst, indicating the specified first day of each week, expressed as tinyint.
@ @dbts	Returns the timestamp of the current database.
@ @error	Returns the error number most recently generated by the system.
@ @errorlog	Returns the full path to the directory in which the Adaptive Server errorlog is kept, relative to \$SYBASE directory (%SYBASE% on NT).
@ @failedoverconn	Returns a value greater than 0 if the connection to the primary companion has failed over and is executing on the secondary companion server. Used only in a high availability environment, and is session-specific.
@ @guestuserid	Returns the ID of the guest user.
@ @hacmpservername	Returns the name of the companion server in a high availability setup.
@ @hacconnection	Returns a value greater than 0 if the connection has the failover property enabled. This is a session-specific property.
@ @heapmemsize	Returns the size of the heap memory pool, in bytes.
@ @identity	Returns the most recently generated IDENTITY column value.
@ @idle	Returns the number of seconds, in CPU time, that Adaptive Server has been idle.
@ @invaliduserid	Returns a value of -1 for an invalid user ID.
@ @io_busy	Returns the number of seconds in CPU time that Adaptive Server has spent doing input and output operations.



---

@ @isolation	Returns the value of the session-specific isolation level (0, 1, or 3) of the current Transact-SQL program.
@ @kernel_addr	Returns the starting address of the first shared memory region that contains the kernel region. The result is in the form of <i>0xaddress pointer value</i> .
@ @kernel_size	Returns the size of the kernel region that is part of the first shared memory region.
@ @langid	Returns the server-wide language ID of the language in use, as specified in <code>syslanguages.langid</code> .
@ @language	Returns the name of the language in use, as specified in <code>syslanguages.name</code> .
@ @lock_timeout	Set using <code>set lock wait n</code> . Returns the current <i>lock_timeout</i> setting, in milliseconds.
@ @maxcharlen	Returns the maximum length, in bytes, of a character in Adaptive Server's default character set.
@ @max_connections	Returns the maximum number of simultaneous connections that can be made with Adaptive Server in the current computer environment. You can configure Adaptive Server for any number of connections less than or equal to the value of @ @max_connections with the number of user connections configuration parameter.
@ @maxgroupid	Returns the highest group user ID. The highest value is 1048576.
@ @maxpagesize	Returns the server's logical page size.
@ @max_precision	Returns the precision level used by decimal and numeric datatypes set by the server. This value is a fixed constant of 38.
@ @maxspid	Returns maximum valid value for the spid.
@ @maxsuid	Returns the highest server user ID. The default value is 2147483647.
@ @maxuserid	Returns the highest user ID. The highest value is 2147483647.
@ @mempool_addr	Returns the global memory pool table address. The result is in the form <i>0xaddress pointer value</i> . This variable is for internal use.
@ @mingroupid	Returns the lowest group user ID. The lowest value is 16384.
@ @min_poolsize	Returns the minimum size of a named cache pool, in kilobytes. It is calculated based on the <code>DEFAULT_POOL_SIZE</code> , which is 256, and the current value of max database page size.
@ @minspid	Returns 1, which is the lowest value for spid.
@ @minsuid	Returns the minimum server user ID. The lowest value is -32768.

<code>@ @userid</code>	Returns the lowest user ID. The lowest value is -32768.
<code>@ @ncharsize</code>	Returns the maximum length, in bytes, of a character set in the current server default character set.
<code>@ @nestlevel</code>	Returns the current nesting level.
<code>@ @nodeid</code>	Returns the current installation's 48-bit node identifier. Adaptive Server generates a nodeid the first time the master device is first used, and uniquely identifies an Adaptive Server installation.
<code>@ @options</code>	Returns a hexadecimal representation of the session's set options.
<code>@ @packet_errors</code>	Returns the number of errors detected by Adaptive Server while reading and writing packets.
<code>@ @pack_received</code>	Returns the number of input packets read by Adaptive Server.
<code>@ @pack_sent</code>	Returns the number of output packets written by Adaptive Server.
<code>@ @pagesize</code>	Returns the server's virtual page size.
<code>@ @parallel_degree</code>	Returns the current maximum parallel degree setting.
<code>@ @probesuid</code>	Returns a value of 2 for the probe user ID.
<code>@ @procid</code>	Returns the stored procedure ID of the currently executing procedure.
<code>@ @rowcount</code>	Returns the number of rows affected by the last query. <code>@ @rowcount</code> is set to 0 by any command that does not return rows, such as an if, update, or delete statement. With cursors, <code>@ @rowcount</code> represents the cumulative number of rows returned from the cursor result set to the client, up to the last fetch request.
<code>@ @scan_parallel_degree</code>	Returns the current maximum parallel degree setting for nonclustered index scans.
<code>@ @servername</code>	Returns the name of Adaptive Server.
<code>@ @shmem_flags</code>	Returns the shared memory region properties. This variable is for internal use. There are a total of 13 different properties values corresponding to 13 bits in the integer. The valid values represented from low to high bit are: MR_SHARED, MR_SPECIAL, MR_PRIVATE, MR_READABLE, MR_WRITABLE, MR_EXECUTABLE, MR_HWCOHERENCY, MR_SWCOHERENC, MR_EXACT, MR_BEST, MR_NAIL, MR_PSUEDO, MR_ZERO.
<code>@ @spid</code>	Returns the server process ID of the current process.
<code>@ @sqlstatus</code>	Returns status information (warning exceptions) resulting from the execution of a fetch statement.

---

@ @stringsize	Returns the amount of character data returned from a toString() method. The default is 50. Max values may be up to 2GB. A value of zero specifies the default value.
@ @tempdbid	Returns a valid temporary database ID (dbid) of the session's assigned temporary database.
@ @textcolid	Returns the column ID of the column referenced by @@textptr.
@ @textdbid	Returns the database ID of a database containing an object with the column referenced by @@textptr.
@ @textobjid	Returns the object ID of an object containing the column referenced by @@textptr.
@ @textptr	Returns the text pointer of the last text or image column inserted or updated by a process (Not the same as the textptr function).
@ @textptr_parameters	Returns 0 if the current status of the textptr_parameters configuration parameter is off. Returns 1 if the current status of the textptr_parameters if on.
@ @textsize	Returns the limit on the number of bytes of text or image data a select returns. Default limit is 32K bytes for isql; the default depends on the client software. Can be changed for a session with set textsize.
@ @textts	Returns the text timestamp of the column referenced by @@textptr.
@ @thresh_hysteresis	Returns the decrease in free space required to activate a threshold. This amount, also known as the hysteresis value, is measured in 2K database pages. It determines how closely thresholds can be placed on a database segment.
@ @timeticks	Returns the number of microseconds per tick. The amount of time per tick is machine-dependent.
@ @total_errors	Returns the number of errors detected by Adaptive Server while reading and writing.
@ @total_read	Returns the number of disk reads by Adaptive Server.
@ @total_write	Returns the number of disk writes by Adaptive Server.
@ @tranchained	Returns 0 if the current transaction mode of the Transact-SQL program is unchained. Returns 1 if the current transaction mode of the Transact-SQL program is chained.
@ @trancount	Returns the nesting level of transactions in the current user session.
@ @transactional_rpc	Returns 0 if RPCs to remote servers are transactional. Returns 1 if RPCs to remote servers are not transactional.

@ @ <i>transtate</i>	Returns the current state of a transaction after a statement executes in the current user session.
@ @ <i>unicharsize</i>	Returns 2, the size of a character in <i>unichar</i> .
@ @ <i>version</i>	Returns the date, version string, and so on of the current release of Adaptive Server.
@ @ <i>version_as_integer</i>	Returns the version of the current release of Adaptive Server as an integer.

## Functions

See *Reference Manual: Building Blocks* for details.

<i>abs</i>	<i>abs(numeric_expression)</i>
<i>acos</i>	<i>acos(cosine)</i>
<i>ascii</i>	<i>ascii(char_expr   uchar_expr)</i>
<i>asin</i>	<i>asin(sine)</i>
<i>atan</i>	<i>atan(tangent)</i>
<i>atn2</i>	<i>atn2(sine, cosine)</i>
<i>avg</i>	<i>avg([all   distinct] expression)</i>
<i>ceiling</i>	<i>ceiling(value)</i>
<i>char</i>	<i>char(integer_expr)</i>
<i>charindex</i>	<i>charindex(expression1, expression2)</i>
<i>char_length</i>	<i>char_length(char_expr   uchar_expr)</i>
<i>col_length</i>	<i>col_length(object_name, column_name)</i>
<i>col_name</i>	<i>col_name(object_id, column_id[, database_id])</i>
<i>compare</i>	<i>compare ({char_expression1 uchar_expression1}, {char_expression2 uchar_expression2}), [{collation_name   collation_ID}]</i>
<i>convert</i>	<i>convert (datatype [(length)   (precision[, scale])], [null   not null], expression [, style])</i>
<i>cos</i>	<i>cos(angle)</i>
<i>cot</i>	<i>cot(angle)</i>
<i>count</i>	<i>count([all   distinct] expression)</i>

---

curunreservedpgs	curunreservedpgs( <i>dbid</i> , <i>lstart</i> , <i>unreservedpgs</i> )
data_pgs	data_pgs( <i>[dbid]</i> , <i>object_id</i> , { <i>data_oam_pg_id</i>   <i>index_oam_pg_id</i> })
datalength	datalength( <i>expression</i> )
dateadd	dateadd( <i>date_part</i> , <i>integer</i> , { <i>date</i>   <i>time</i>   <i>datetime</i> })
datediff	datediff( <i>datepart</i> , { <i>date1</i> , <i>date2</i>   <i>time1</i>   <i>time2</i>   <i>datetime1</i>   <i>datetime2</i> })
datename	datename ( <i>datepart</i> , { <i>date</i>   <i>time</i>   <i>datetime</i> })
datepart	datepart( <i>date_part</i> , { <i>date</i>   <i>time</i>   <i>datetime</i> })
day	day( <i>date_expression</i> )
db_id	db_id( <i>database_name</i> )
db_name	db_name( <i>[database_id]</i> )
degrees	degrees( <i>numeric</i> )
derived_stat	derived_stat({ <i>object_name</i>   <i>object_id</i> }, { <i>index_name</i>   <i>index_id</i> }, "statistic")
difference	difference( <i>expr1</i> , <i>expr2</i> )
exp	exp( <i>approx_numeric</i> )
floor	floor( <i>numeric</i> )
get_appcontext	get_appcontext ("context_name", "attribute_name")
getdate	getdate()
hextoint	hextoint ( <i>hexadecimal_string</i> )
host_id	host_id()
host_name	host_name()
identity_burn_max	identity_burn_max( <i>table_name</i> )
index_col	index_col ( <i>object_name</i> , <i>index_id</i> , <i>key_#</i> [, <i>user_id</i> ])
index_colorder	index_colorder ( <i>object_name</i> , <i>index_id</i> , <i>key_#</i> [, <i>user_id</i> ])
inttohex	inttohex ( <i>integer_expression</i> )
isnull	isnull( <i>expression1</i> , <i>expression2</i> )
is_sec_service_on	is_sec_service_on( <i>security_service_nm</i> )
lct_admin	lct_admin({{"lastchance"   "logfull"   "reserved_for_rollbacks"}, <i>database_id</i>  "reserve", { <i>log_pages</i>   0 }  "abort", <i>process-id</i> [, <i>database-id</i> ])
left	left( <i>character_expression</i> , <i>integer_expression</i> )
len	len( <i>string_expression</i> )

license_enabled	license_enabled("ase_server"   "ase_ha"   "ase_dtm"   "ase_java"   "ase_asm")
list_appcontext	list_appcontext(["context_name"])
lockscheme	lockscheme( <i>object_name</i> )
	Or
	lockscheme( <i>object_id</i> [, <i>db_id</i> ])
log	log( <i>approx_numeric</i> )
log10	log10( <i>approx_numeric</i> )
lower	lower( <i>char_expr</i>   <i>uchar_expr</i> )
ltrim	ltrim( <i>char_expr</i>   <i>uchar_expr</i> )
max	max( <i>expression</i> )
min	min( <i>expression</i> )
month	month( <i>date_expression</i> )
mut_excl_roles	mut_excl_roles ( <i>role1</i> , <i>role2</i> [membership   activation])
newid	newid([ <i>optionflag</i> ])
next_identity	next_identity( <i>table_name</i> )
object_id	object_id( <i>object_name</i> )
object_name	object_name( <i>object_id</i> , <i>database_id</i> )
pagesize	pagesize( <i>object_name</i> [, <i>index_name</i> ]) or, pagesize( <i>object_id</i> [, <i>db_id</i> [, <i>index_id</i> ]])
patindex	patindex("% <i>pattern</i> %", <i>char_expr</i>   <i>uchar_expr</i> [, using {bytes   characters   chars} ] )
pi	pi()
power	power( <i>value</i> , <i>power</i> )
proc_role	proc_role ("role_name")
ptn_data_pgs	ptn_data_pgs( <i>object_id</i> , <i>partition_id</i> )
radians	radians( <i>numeric</i> )
rand	rand([ <i>integer</i> ])
replicate	replicate ( <i>char_expr</i>   <i>uchar_expr</i> , <i>integer_expr</i> )
reserved_pgs	reserved_pgs( <i>object_id</i> , {doampg   ioampg})
reverse	reverse( <i>expression</i>   <i>uchar_expr</i> )
right	right( <i>expression</i> , <i>integer_expr</i> )

---

rm_appcontext	rm_appcontext ("context_name", "attribute_name")
role_contain	role_contain("role1", "role2")
role_id	role_id("role_name")
role_name	role_name(role_id)
round	round(number, decimal_places)
rowcnt	rowcnt(sysindexes.doampg)
rtrim	rtrim(char_expr   uchar_expr)
set_appcontext	set_appcontext ("context_name", "attribute_name", "attribute_value")
show_role	show_role()
show_sec_services	show_sec_services()
sign	sign(numeric)
sin	sin(approx_numeric)
sortkey	sortkey (char_expression   uchar_expression) [, {collation_name   collation_ID}]
soundex	soundex(char_expr   uchar_expr)
space	space(integer_expr)
square	square(numeric_expression)
sqrt	sqrt(approx_numeric)
str	str(approx_numeric [, length [, decimal] ])
str_replace	replace("string_expression1", "string_expression2", "string_expression3")
stuff	stuff(char_expr1   uchar_expr1, start, length, char_expr2   uchar_expr2)
substring	substring(expression, start, length )
sum	sum([all   distinct] expression)
suser_id	suser_id([server_user_name])
suser_name	suser_name([server_user_id])
syb_quit()	syb_quit()
syb_sendmsg	syb_sendmsg ip_address, port_number, message
tan	tan(angle)
tempdb_id	tempdb_id()
textptr	textptr(column_name)
textvalid	textvalid("table_name.column_name", textpointer)
to_unichar	to_unichar (integer_expr)

tsequal	tsequal( <i>browsed_row_timestamp</i> , <i>stored_row_timestamp</i> )
uhighsurr	uhighsurr( <i>uchar_expr</i> , start)
ulowsurr	ulowsurr( <i>uchar_expr</i> , start)
upper	upper( <i>char_expr</i> )
uscalar	uscalar( <i>uchar_expr</i> )
used_pgs	used_pgs( <i>object_id</i> , <i>doampg</i> , <i>ioampg</i> )
user	user
user_id	user_id( <i>[user_name]</i> )
user_name	user_name( <i>[user_id]</i> )
valid_name	valid_name( <i>character_expression</i> )
valid_user	valid_user( <i>server_user_id</i> )
year	year( <i>date_expression</i> )

## Commands

See *Reference Manual: Commands* for details.

alter database	<pre>alter database <i>database_name</i>   [on {default   <i>database_device</i> } [= size]    [, <i>database_device</i> [= size]]...]   [log on { default   <i>database_device</i> } [= size]    [, <i>database_device</i> [= size]]...]   [with override]   [for load]   [for proxy_update]</pre>
alter role	<pre>alter role <i>role1</i> { add   drop } exclusive { membership   activation } <i>role2</i> alter role <i>role_name</i> [add passwd "<i>password</i>"     drop passwd] [lock   unlock] alter role { <i>role_name</i>   "all overrides" }   set { passwd expiration   min passwd length     max failed_logins } <i>option_value</i></pre>
alter table	<pre>alter table <i>[[database.][owner].table_name</i>   { add <i>column_name datatype</i>     [default {<i>constant_expression</i>   user   null}]     {identity   null   not null}     [off row   in row]     [ [constraint <i>constraint_name</i>]</pre>



---

```

    { { unique | primary key }
      [clustered | nonclustered]
      [asc | desc]
      [with { fillfactor = pct,
             max_rows_per_page = num_rows,
             reservepagegap = num_pages }]
      [on segment_name]
    | references [[database.]owner.]ref_table
      [(ref_column)]
    | check (search_condition) ] ... }
  [, next_column]...

| add {[constraint constraint_name]
{ unique | primary key }
  [clustered | nonclustered]
  (column_name [asc | desc]
  [, column_name [asc | desc]...])
  [with { fillfactor = pct,
         max_rows_per_page = num_rows,
         reservepagegap = num_pages}]
  [on segment_name]
| foreign key (column_name [{, column_name}...])
  references [[database.]owner.]ref_table
  [(ref_column [{, ref_column}...])]
  [
| check (search_condition)}

| drop {column_name [, column_name]...
  | constraint constraint_name }

| modify column_name datatype [null | not null]
  [, next_column]...

| replace column_name
  default { constant_expression | user | null}

| partition number_of_partitions

| unpartition

| { enable | disable } trigger

| lock {allpages | datarows | datapages } }

| with exp_row_size=num_bytes

begin...end      begin
                  statement block
end
begin transaction  begin tran[saction] [transaction_name]

```

break	while <i>logical_expression</i> <i>statement</i> break <i>statement</i> continue
case	case when <i>search_condition</i> then <i>expression</i> [when <i>search_condition</i> then <i>expression</i> ]... [else <i>expression</i> ] end case and values syntax: case <i>expression</i> when <i>expression</i> then <i>expression</i> [when <i>expression</i> then <i>expression</i> ]... [else <i>expression</i> ] end
checkpoint	checkpoint [all   [ <i>dbname</i> [, <i>dbname</i> , <i>dbname</i> , .....]]
close	close <i>cursor_name</i>
coalesce	coalesce( <i>expression</i> , <i>expression</i> [, <i>expression</i> ]...)
commit	commit [tran   transaction   work] [ <i>transaction_name</i> ]
compute clause	<i>start_of_select_statement</i> compute <i>row_aggregate</i> ( <i>column_name</i> ) [, <i>row_aggregate</i> ( <i>column_name</i> )]... [by <i>column_name</i> [, <i>column_name</i> ]...]
connect to...disconnect	connect to <i>server_name</i> disconnect
continue	while <i>boolean_expression</i> <i>statement</i> break <i>statement</i> continue
create database	create [temporary] database <i>database_name</i> [on {default   <i>database_device</i> } [= <i>size</i> ] [, <i>database_device</i> [= <i>size</i> ]...] [log on <i>database_device</i> [= <i>size</i> ] [, <i>database_device</i> [= <i>size</i> ]...] [with {override   default_location = "pathname"}] [for {load   proxy_update}]
create default	create default [ <i>owner.</i> ] <i>default_name</i> as <i>constant_expression</i>
create existing table	create existing table <i>table_name</i> ( <i>column_list</i> ) [ on <i>segment_name</i> ]

---

	[ [ external {table   procedure   file} ] at <i>pathname</i> [column delimiter " <i>string</i> "]]
create function (SQLJ)	<pre> create function [owner.]sql_function_name   ( [ sql_parameter_name sql_datatype     [( length)   (precision[, scale ] ) ]     [ [, sql_parameter_name sql_datatype       [( length)   (precision[, scale ] ) ]       ... ] ] )   returns sql_datatype     [ ( length)   (precision[, scale ] ) ]   [modifies sql data]   [returns null on null input       called on null input]   [deterministic   not deterministic]   [exportable]   language java   parameter style java   external name 'java_method_name     [ ( [java_datatype[, java_datatype       ...] ] ) ]'</pre>
create index	<pre> create [unique] [clustered   nonclustered]   index index_name   on [[database.]owner.]table_name     (column_name [asc   desc]     [, column_name [asc   desc]]...)   [with { fillfactor = pct,     max_rows_per_page = num_rows,     reservepagegap = num_pages,     consumers = x, ignore_dup_key, sorted_data,     [ignore_dup_row   allow_dup_row],     statistics using num_steps values } ]   [on segment_name]</pre>
create plan	<pre> create plan query_plan   [into group_name]   [and set @new_id]</pre>
create procedure	<pre> create procedure [owner.]procedure_name[;number]   [( [ @parameter_name     datatype [(length)   (precision [, scale ])]     [= default][output]     [, @parameter_name     datatype [(length)   (precision [, scale ])]     [= default][output]... ] ] ]   [with recompile]   as {SQL_statements   external name dll_name}</pre>
create procedure (SQLJ)	<pre> create procedure [owner.]sql_procedure_name   ([ [ in   out   inout ] sql_parameter_name     sql_datatype [( length)       (precision[, scale] ) ]</pre>

```

        [, [ in | out | inout ] sql_parameter_name
           sql_datatype [( length ) |
                        ( precision [, scale ] ) ] ]
        ...)
    [modifies sql data ]
    [dynamic result sets integer]
    [deterministic | not deterministic]
    language java
    parameter style java
    external name 'java_method_name
        [ ( [java_datatype [, java_datatype
        ...] ) ] )'
```

create proxy\_table            create proxy\_table *table\_name*  
                                   [on *segment\_name*]  
                                   [ external [ table | directory | file ] ]  
                                   at *pathname*  
                                   [column delimiter "<string>"]

create role                    create role *role\_name* [ with passwd "*password*"  
                                   [, { "passwd expiration" | "min passwd length" |  
                                   "max failed\_logins" } *option\_value* ] ]

create rule                    create [ [ and | or ] access]] rule  
                                   [*owner*.]*rule\_name*  
                                   as *condition\_expression*

create schema                 create schema authorization *authorization\_name*  
                                   *create\_object\_statement*  
                                   [*create\_object\_statement* ... ]  
                                   [*permission\_statement* ... ]

create table                  create table [*database* .]*owner* .]*table\_name* (*column\_name datatype*  
                                   [default {*constant\_expression* | user | null}]  
                                   {{[identity | null | not null]}  
                                   [off row | [ in row [ (*size\_in\_bytes*) ] ]  
                                   [[constraint *constraint\_name* ]  
                                   {{unique | primary key}  
                                   [clustered | nonclustered] [asc | desc]  
                                   [with { fillfactor = *pct*,  
                                   max\_rows\_per\_page = *num\_rows*, }  
                                   reservepagegap = *num\_pages* }]  
                                   [on *segment\_name*]  
                                   | references [[*database* .]*owner* .]*ref\_table*  
                                   [(*ref\_column* )]  
                                   | check (*search\_condition*)]})...  
                                   | [constraint *constraint\_name*]  
                                   {{unique | primary key}  
                                   [clustered | nonclustered]  
                                   (*column\_name* [asc | desc]  
                                   [ { , *column\_name* [asc | desc] } ... ] )  
                                   [with { fillfactor = *pct*  
                                   max\_rows\_per\_page = *num\_rows* ,

---

```

        reservepagegap = num_pages } ]
    [on segment_name]
    |foreign key (column_name [{,column_name}...])
    references [(database.owner).ref_table
        [(ref_column [{, ref_column}...])]
    | check (search_condition) ... }
    [{, {next_column | next_constraint}...}]
    [lock {datarows | datapages | allpages } ]
    [with { max_rows_per_page = num_rows,
        exp_row_size = num_bytes,
        reservepagegap = num_pages,
        identity_gap = value } ]
    [on segment_name ]
    [ [ external table ] at pathname ]

```

```

create trigger
    create trigger [owner .]trigger_name
    on [owner .]table_name
    for {insert , update , delete}
    as SQL_statements

```

Or, using the if update clause:

```

create trigger [owner .]trigger_name
    on [owner .]table_name
    for {insert , update}
    as
        [if update (column_name )
            [{and | or} update (column_name )]...
            SQL_statements
        [if update (column_name )
            [{and | or} update (column_name )]...
            SQL_statements ]...

```

```

create view
    create view [owner .]view_name
    [(column_name [, column_name ]...)]
    as
    select [distinct] select_statement
    [with check option]

```

```

dbcc
    dbcc addtempdb( dbid | databbase_name )
    dbcc checkalloc [(database_name [, fix | nofix])]
    dbcc checkcatalog [(database_name)]
    dbcc checkdb [(database_name [, skip_ncindex])]
    dbcc checkstorage [(database_name)]
    dbcc checktable({table_name | table_id}[ , skip_ncindex])
    dbcc checkverify [(database_name)]
    dbcc complete_xact (xid, {"commit" | "rollback"})
    dbcc forget_xact (xid)

```

```

dbcc dbrepair (database_name, dropdb)
dbcc engine( {offline , [enginenum] | "online" })
dbcc fix_text ({table_name | table_id})
dbcc indexalloc ({table_name | table_id}, index_id
    [, {full | optimized | fast | null}
    [, fix | nofix]])
dbcc pravailabletempdbs
dbcc rebuild_text (table [, column [, text_page_number]])
dbcc reindex ({table_name | table_id})
dbcc tablealloc ({table_name | table_id}
    [, {full | optimized | fast | null}
    [, fix | nofix]])
dbcc { traceon | traceoff } (flag [, flag ... ])
dbcc tune ( { ascinserts, {0 | 1 } , tablename |
    cleanup, {0 | 1 } |
    cpuaffinity, start_cpu {, on| off } |
    des_greedyalloc, dbid, object_name,
    " { on | off }" |
    deviochar vdevno, "batch_size" |
    doneinproc { 0 | 1 } |
    maxwritedes, writes_per_batch })

deallocate cursor          deallocate cursor cursor_name

declare                    Variable declaration:
                            declare @variable_name datatype [, @variable_name datatype]...

                            Variable assignment:
select @variable = {expression | select_statement}
    [, @variable = {expression | select_statement} ...]
    [from table_list]
    [where search_conditions]
    [group by group_by_list]
    [having search_conditions]
    [order by order_by_list]
    [compute function_list [by by_list]]

declare cursor             declare cursor_name cursor
                            for select_statement
                            [for {read only | update [of column_name_list]}]

delete                     delete [from]
                            [[database.]owner.]{view_name|table_name}
                            [where search_conditions]
                            [plan "abstract plan"]

delete                    delete [[database.]owner.]{table_name | view_name}
                            [from [[database.]owner.]{view_name [readpast]}]

```

---

```

        table_name [readpast]
            [(index {index_name | table_name }
              [ prefetch_size ][lru|mru])}]
    [, [[database.]owner.]{view_name [readpast]
      table_name [readpast]
        [(index {index_name | table_name }
          [ prefetch_size ][lru|mru])}] ...]
    [where search_conditions ]
    [plan "abstract plan"]
delete [from] [[database.]owner.]{table_name|view_name}
  where current of cursor_name
delete statistics table_name [(column_name
  [, column_name]...)]
disk init
  disk init
    name = "device_name" ,
    physname = "physicalname" ,
    [vdevno = virtual_device_number ,]
    size = number_of_blocks
    [, vstart = virtual_address
      , cntrltype = controller_number ]
    [, contiguous]
    [, dsync = { true | false } ]
disk mirror
  disk mirror
    name = "device_name" ,
    mirror = "physicalname"
    [, writes = { serial | noserial } ]
disk refit
  disk refit
disk reinit
  disk reinit
    name = "device_name",
    physname = "physicalname" ,
    [vdevno = virtual_device_number ,]
    size = number_of_blocks
    [, vstart = virtual_address
      , cntrltype = controller_number]
    [, dsync = { true | false } ]
disk remirror
  disk remirror
    name = "device_name"
disk resize
  disk resize
    name = "device_name",
    size = additional_space
disk unmirror
  disk unmirror
    name = "device_name"
    [,side = { "primary" | secondary } ]
    [,mode = { retain | remove } ]
drop database
  drop database database_name [, database_name] ...

```

```

drop default                drop default [owner.]default_name [, [owner.]default_name] ...
drop function (SQLJ)       drop func[tion] [owner.]function_name[, [owner.]function_name] ...
drop index                 drop index table_name.index_name [, table_name.index_name] ...
drop procedure            drop proc[edure] [owner.]procedure_name
                           [, [owner.]procedure_name] ...
drop role                 drop role role_name [with override]
drop rule                 drop rule [owner.]rule_name [, [owner.]rule_name] ...
drop table               drop table [[database.]owner.]table_name
                           [, [[database.]owner.]table_name ] ...
drop trigger             drop trigger [owner.]trigger_name [, [owner.]trigger_name] ...
drop view                drop view [owner.]view_name [, [owner.]view_name] ...
dump database            dump database database_name
                           to [compress::[compression_level::]]stripe_device
                           [at backup_server_name]
                           [density = density_value,
                           blocksize = number_bytes,
                           capacity = number_kilobytes,
                           dumpvolume = volume_name,
                           file = file_name]
                           [stripe on [compress::[compression_level::]]stripe_device
                           [at backup_server_name]
                           [density = density_value,
                           blocksize = number_bytes,
                           capacity = number_kilobytes,
                           dumpvolume = volume_name,
                           file = file_name]]
                           [[stripe on [compress::[compression_level::]]stripe_device
                           [at backup_server_name]
                           [density = density_value,
                           blocksize = number_bytes,
                           capacity = number_kilobytes,
                           dumpvolume = volume_name,
                           file = file_name]]...]
                           [with {
                           density = density_value,
                           blocksize = number_bytes,
                           capacity = number_kilobytes,
                           dumpvolume = volume_name,
                           file = file_name,
                           [dismount | nodismount],
                           [nounload | unload],
                           retaindays = number_days,
                           [noinit | init],
                           notify = {client | operator_console}
                           } ]

```



---

dump transaction

To make a routine log dump:

```
dump tran[saction] database_name
to [compress::[compression_level::]]stripe_device
[at backup_server_name]
[density = density_value,
blocksize = number_bytes,
capacity = number_kilobytes,
dumpvolume = volume_name,
file = file_name]
[[stripe on [compress::[compression_level::]]stripe_device
[at backup_server_name]
[density = density_value,
blocksize = number_bytes,
capacity = number_kilobytes,
dumpvolume = volume_name,
file = file_name]]
[[stripe on [compress::[compression_level::]]stripe_device
[at backup_server_name]
[density = density_value,
blocksize = number_bytes,
capacity = number_kilobytes,
dumpvolume = volume_name,
file = file_name] ...]
[with {
density = density_value,
blocksize = number_bytes,
capacity = number_kilobytes,
dumpvolume = volume_name,
file = file_name,
[dismount | nodismount],
[nounload | unload],
retaindays = number_days,
[noinit | init],
notify = {client | operator_console},
standby_access }
```

To truncate the log without making a backup copy:

```
dump tran[saction] database_name
with truncate_only
```

To truncate a log that is filled to capacity. *Use only as a last resort:*

```
dump tran[saction] database_name
with no_log
```

To back up the log after a database device fails:

```
dump tran[saction] database_name
to [compress::[compression_level::]]stripe_device
[at backup_server_name]
[density = density_value,
```

```

        blocksize = number_bytes,
        capacity = number_kilobytes,
        dumpvolume = volume_name,
        file = file_name]
[stripe on [compress::compression_level::] stripe_device
 [at backup_server_name]
 [density = density_value,
 blocksize = number_bytes,
 capacity = number_kilobytes,
 dumpvolume = volume_name,
 file = file_name]]
[[stripe on [compress::compression_level::] stripe_device
 [at backup_server_name]
 [density = density_value,
 blocksize = number_bytes,
 capacity = number_kilobytes,
 dumpvolume = volume_name,
 file = file_name] ]...]
[with {
 density = density_value,
 blocksize = number_bytes,
 capacity = number_kilobytes,
 dumpvolume = volume_name,
 file = file_name,
 [dismount | nodismount],
 [nounload | unload],
 retaindays = number_days,
 [noinit | init],
 no_truncate,
 notify = {client | operator_console}}]
execute [exec[ute]] [ @return_status = ]
        [[[server .]database.]owner.]procedure_name[;number]
        [[ @parameter_name =] value |
         [ @parameter_name =] @variable [output]
        [, [ @parameter_name =] value |
         [ @parameter_name =] @variable [output]...]]
        [with recompile]
or
exec[ute] ("string" | char_variable [+ "string" | char_variable]...)
fetch fetch_cursor_name [ into fetch_target_list ]
goto label label:
        goto label
grant To grant permission to access database objects:
grant {all [privileges]} permission_list}
on { table_name [(column_list)]
    | view_name[(column_list)]

```

---

```
    | stored_procedure_name}
to {public | name_list | role_name}
[with grant option]
```

To grant permission to execute certain commands:

```
grant {all [privileges] | command_list}
to {public | name_list | role_name}
```

To grant a role to a user or a role:

```
grant {role role_granted [, role_granted ...]}
to grantee [, grantee...]
```

To grant and revoke access on certain dbcc commands:

```
grant dbcc {dbcc_command [on {all | database }
[, dbcc_command [on {all | database }], ...]}
to { user_list | role_list }
```

group by and having  
clauses

```
Start of select statement
[group by [all] aggregate_free_expression
[, aggregate_free_expression]...]
[having search_conditions]
End of select statement
```

if...else

```
if logical_expression [plan "abstract plan"]
statements
[else
[if logical_expression] [plan "abstract plan"]
statement]
```

insert

```
insert [into] [database.owner.]{table_name|view_name}
[(column_list)]
{values (expression [, expression]...)
|select_statement [plan "abstract plan"] }
```

kill

```
kill spid
```

load database

```
load database database_name
from [compress::]stripe_device
[at backup_server_name ]
[density = density_value,
blocksize = number_bytes,
dumpvolume = volume_name,
file = file_name]
[stripe on [compress::]stripe_device
[at backup_server_name ]
[density = density_value,
blocksize = number_bytes,
dumpvolume = volume_name,
file = file_name]
[[stripe on [compress::]stripe_device
```

```

[at backup_server_name ]
[density = density_value,
blocksize = number_bytes,
dumpvolume = volume_name,
file = file_name]...]
[with {
density = density_value,
blocksize = number_bytes,
dumpvolume = volume_name,
file = file_name,
[dismount | nodismount],
[nounload | unload],
listonly [= full],
headeronly,
notify = {client | operator_console}
}}]
load transaction load tran[saction] database_name
from [compress::]stripe_device
[at backup_server_name ]
[density = density_value,
blocksize = number_bytes,
dumpvolume = volume_name,
file = file_name]
[stripe on [compress::]stripe_device
[at backup_server_name ]
[density = density_value,
blocksize = number_bytes,
dumpvolume = volume_name,
file = file_name]
[[stripe on [compress::]stripe_device
[at backup_server_name ]
[density = density_value,
blocksize = number_bytes,
dumpvolume = volume_name,
file = file_name]...]
[with {
density = density_value,
blocksize = number_bytes,
dumpvolume = volume_name,
file = file_name,
[dismount | nodismount],
[nounload | unload],
listonly [= full],
headeronly,
notify = {client | operator_console}
until_time = datetime}]]
lock table lock table table_name in {share | exclusive } mode
[ wait [ numsecs ] | nowait ]
mount mount all from <manifest_file>

```

---

	mount database all from <manifest_file> with listonly
nullif	nullif( <i>expression</i> , <i>expression</i> )
online database	online database <i>database_name</i> [for standby_access]
open	open <i>cursor_name</i>
order by clause	[ <i>Start of select statement</i> ] [order by {[ <i>table_name</i> .  <i>view_name</i> .] <i>column_name</i>   <i>select_list_number</i>   <i>expression</i> } [asc   desc] [, {[ <i>table_name</i> .  <i>view_name</i> .] <i>column_name</i> <i>select_list_number</i> { <i>expression</i> } [asc   desc]}...] [ <i>End of select statement</i> ]
prepare transaction	prepare tran[saction]
print	print { <i>format_string</i>   @ <i>local_variable</i>   @ <i>global_variable</i> } [, <i>arg_list</i> ]
quiesce database	quiesce database <i>tag_name</i> hold <i>dbname</i> [, <i>dbname</i> ] ... [for external dump] [ <i>manifest_file</i> ]
	or:
	quiesce database <i>tag_name</i> release
raiserror	raiserror <i>error_number</i> [ <i>format_string</i>   @ <i>local_variable</i> ] [, <i>arg_list</i> ] [with errordata <i>restricted_select_list</i> ]
readtext	readtext [[ <i>database</i> .] <i>owner</i> .] <i>table_name.column_name</i> <i>text_pointer</i> offset size [holdlock   noholdlock] [readpast] [using {bytes   chars   characters}] [at isolation { [ read uncommitted   0 ]   [ read committed   1 ]   [ repeatable read   2 ]   [ serializable   3 ] } ]
reconfigure	reconfigure
remove java	remove java class <i>class_name</i> [, <i>class_name</i> ]...   package <i>package_name</i> [, <i>package_name</i> ]...   jar <i>jar_name</i> [, <i>jar_name</i> ]...[retain classes]
reorg	reorg reclaim_space <i>tablename</i> [ <i>indexname</i> ] [with {resume, time = <i>no_of_minutes</i> }] reorg forwarded_rows <i>tablename</i> [with {resume,time = <i>no_of_minutes</i> }]

	reorg compact <i>tablename</i> [with {resume, time = <i>no_of_minutes</i> }]
	reorg rebuild <i>tablename</i> [ <i>indexname</i> ]
return	return [ <i>integer_expression</i> ] [plan "abstract plan"]
revoke	To revoke permission to access database objects: revoke [grant option for] {all [privileges]   <i>permission_list</i> } on { <i>table_name</i> [( <i>column_list</i> )]   <i>view_name</i> [( <i>column_list</i> )]   <i>stored_procedure_name</i> } from {public   <i>name_list</i>   <i>role_name</i> } [cascade]
	To revoke permission to create database objects, execute set proxy, or execute set session authorization: revoke {all [privileges]   <i>command_list</i> } from {public   <i>name_list</i>   <i>role_name</i> }
	To revoke a role from a user or another role: revoke role { <i>role_name</i> [, <i>role_name</i> ...]} from { <i>grantee</i> [, <i>grantee</i> ...]}
	To revoke access on some dbcc commands. revoke dbcc { <i>dbcc_command</i> [on {all   <i>database</i> }] [, <i>dbcc_command</i> [on {all   <i>database</i> }], ...]} from { <i>user_list</i>   <i>role_list</i> }
rollback	rollback [tran   transaction   work] [ <i>transaction_name</i>   <i>savepoint_name</i> ]
rollback trigger	rollback trigger [with <i>raiserror_statement</i> ]
save transaction	save transaction <i>savepoint_name</i>
select	select ::= select [ all   distinct ] <i>select_list</i> [ <i>into_clause</i> ] [ <i>from_clause</i> ] [ <i>where_clause</i> ] [ <i>group_by_clause</i> ] [ <i>having_clause</i> ] [ <i>order_by_clause</i> ] [ <i>compute_clause</i> ] [ <i>read_only_clause</i> ] [ <i>isolation_clause</i> ] [ <i>browse_clause</i> ] [ <i>plan_clause</i> ]

---

*select\_list* ::=

---

**Note** For details on *select\_list*, see the parameters description.

---

*into\_clause* ::=

into [[*database*.]*owner*.]*table\_name*  
[ lock {*datarows* | *datapages* | *allpages* } ]  
[ with *into\_option* [, *into\_option*] ...]

*into\_option* ::=

| max\_rows\_per\_page = *num\_rows*  
| exp\_row\_size = *num\_bytes*  
| reservepagegap = *num\_pages*  
| identity\_gap = *gap*  
[existing table *table\_name*]  
[[*external type*] at "*path\_name*"  
[column delimiter *delimiter*]]

*from\_clause* ::=

from *table\_reference* [, *table\_reference*]...

*table\_reference* ::=

*table\_view\_name* | *ANSI\_join*

*table\_view\_name* ::=

[[*database*.]*owner*.] {{*table\_name* | *view\_name*}}  
[as] [*correlation\_name*]  
[index {*index\_name* | *table\_name* }]  
[parallel [*degree\_of\_parallelism*]]  
[prefetch size ][*lru* | *mru*]}  
[holdlock | noholdlock]  
[readpast]  
[shared]

*ANSI\_join* ::=

*table\_reference* *join\_type* join *table\_reference*  
*join\_conditions*  
*join\_type* ::= inner | left [outer] | right [outer]  
*join\_conditions* ::= on *search\_conditions*

*where\_clause* ::=

where *search\_conditions*

*group\_by\_clause* ::=

group by [all] *aggregate\_free\_expression*  
[, *aggregate\_free\_expression*]...

*having\_clause* ::=

having *search\_conditions*

*order\_by\_clause* ::=

order by *sort\_clause* [, *sort\_clause*]...

```

    sort_clause ::=
        { [[[database.]owner.]{table_name.|view_name.}]column_name
          | select_list_number
          | expression }
          [asc | desc]

    compute_clause ::=
        compute row_aggregate(column_name)
            [, row_aggregate(column_name)]...
            [by column_name [, column_name]...]

    read_only_clause ::=
        for {read only | update [of column_name_list]}

    isolation_clause ::=
        at isolation
            { read uncommitted | 0 }
            | { read committed | 1 }
            | { repeatable read | 2 }
            | { serializable | 3 }

    browse_clause ::=
        for browse

    plan_clause ::=
        plan "abstract plan"

set @variable = expression [, @variable = expression...]
set ansinull {on | off}
set ansi_permissions {on | off}
set arithabort [arith_overflow | numeric_truncation] {on | off}
set arithignore [arith_overflow] {on | off}
set bulk array size number
set bulk batch size number
set {chained, close on endtran, nocount, noexec, parseonly, procid,
    self_recursion, showplan, sort_resources} {on | off}
set char_convert {off | on [with {error | no_error}]} |
    charset [with {error | no_error}]
set cis_rpc_handling {on | off}
set [clientname client_name | clienthostname
    host_name | clientapplname application_name]
set cursor rows number for cursor_name
set {datefirst number, dateformat format, language language}
set explicit_transaction_required [true | false]
set fipsflagger {on | off}
set flushmessage {on | off}

```



---

```

set forceplan {on | off}
set identity_insert [database.owner.]table_name {on | off}
set identity_update table_name {on | off}
set jtc {on | off}
set lock { wait [ numsecs ] | nowait }
set offsets {select, from, order, compute, table,
             procedure, statement, param, execute} {on | off}
set parallel_degree number
set plan {dump | load } [group_name] {on | off}
set plan exists check {on | off}
set plan replace {on | off}
set prefetch [on|off]
set process_limit_action {abort | quiet | warning}
set proxy login_name
set quoted_identifier {on | off}
set role {"sa_role" | "sso_role" | "oper_role" |
         role_name [with passwd "password"]} {on | off}
set {rowcount number, textsize number}
set scan_parallel_degree number
set session authorization login_name
set sort_merge {on | off}
set statistics {io, subquerycache, time} {on | off}
set statistics simulate { on | off }
set strict_dtm_enforcement {on | off}
set string_rtruncation {on | off}
set table count number
set textsize {number}
set transaction isolation level {
    [ read uncommitted | 0 ] |
    [ read committed | 1 ] |
    [ repeatable read | 2 ] |
    [ serializable | 3 ] }
set transactional_rpc {on | off}
setuser ["user_name"]
shutdown [srvname] [with {wait | nowait}]
truncate table [database.]owner.table_name

```

union operator	<pre> select <i>select_list</i> [into <i>clause</i>]       [from <i>clause</i>] [where <i>clause</i>]       [group by <i>clause</i>] [having <i>clause</i>] [union [all]       select <i>select_list</i>       [from <i>clause</i>] [where <i>clause</i>]       [group by <i>clause</i>] [having <i>clause</i>] ]... [order by <i>clause</i>] [compute <i>clause</i>] </pre>
unmount	<pre> unmount database &lt;<i>dbname list</i>&gt; to &lt;<i>manifest_file</i>&gt; </pre>
update	<pre> update [[<i>database.</i>]<i>owner.</i>]{<i>table_name</i>   <i>view_name</i>} set [[(<i>database.</i>]<i>owner.</i>){<i>table_name</i>. <i>view_name.</i>}     <i>column_name1</i> =         {<i>expression1</i>   NULL   (<i>select_statement</i>)}       <i>variable_name1</i> =         {<i>expression1</i>   NULL   (<i>select_statement</i>)} [, <i>column_name2</i> =     {<i>expression2</i>   NULL   (<i>select_statement</i>)}]...   [, <i>variable_name2</i> =     {<i>expression2</i>   NULL   (<i>select_statement</i>)}]...  [from [[<i>database.</i>]<i>owner.</i>]{<i>view_name</i> [readpast]]      <i>table_name</i> [readpast]      [(index {<i>index_name</i>   <i>table_name</i>}       [ prefetch size ][<i>lru</i> <i>mru</i>})]      [, [[<i>database.</i>]<i>owner.</i>]{<i>view_name</i> [readpast]]       <i>table_name</i> [readpast]       [(index {<i>index_name</i>   <i>table_name</i>}        [ prefetch size ][<i>lru</i> <i>mru</i>})]      ]      ] [where <i>search_conditions</i>] [plan "<i>abstract plan</i>"]  update [[<i>database.</i>]<i>owner.</i>]{<i>table_name</i>   <i>view_name</i>} set [[(<i>database.</i>]<i>owner.</i>){<i>table_name</i>. <i>view_name.</i>}     <i>column_name1</i> =         {<i>expression1</i>   NULL   (<i>select_statement</i>)}       <i>variable_name1</i> =         {<i>expression1</i>   NULL   (<i>select_statement</i>)} [, <i>column_name2</i> =     {<i>expression2</i>   NULL   (<i>select_statement</i>)}]...   [, <i>variable_name2</i> =     {<i>expression2</i>   NULL   (<i>select_statement</i>)}]... where current of <i>cursor_name</i> </pre>
update all statistics	<pre> update all statistics <i>table_name</i> </pre>
update partition statistics	<pre> update partition statistics <i>table_name</i> [<i>partition_number</i>] </pre>

---

update statistics	<pre>update statistics <i>table_name</i> [ [<i>index_name</i>]   [( <i>column_list</i> ) ] ]     [using <i>step</i> values]     [with consumers = <i>consumers</i> ]  update index statistics <i>table_name</i> [<i>index_name</i>]     [using <i>step</i> values]     [with consumers = <i>consumers</i> ]</pre>
use	use <i>database_name</i>
waitfor	waitfor { delay <i>time</i>   time <i>time</i>   errorexit   processexit   mirrorexit }
where clause	<p>Search conditions immediately follow the keyword where in a select, insert, update, or delete statement. If you use more than one search condition in a single statement, connect the conditions with and or or.</p> <pre>where [not] <i>expression comparison_operator expression</i> where [not] <i>expression</i> [not] like "<i>match_string</i>"     [escape "<i>escape_character</i> "] where [not] <i>expression</i> is [not] null where [not] <i>expression</i> [not] between <i>expression</i> and <i>expression</i> where [not] <i>expression</i> [not] in ((<i>value_list</i>   <i>subquery</i>)) where [not] exists (<i>subquery</i>) where [not]     <i>expression comparison_operator</i>     {any   all} (<i>subquery</i>) where [not] <i>column_name join_operator column_name</i> where [not] <i>logical_expression</i> where [not] <i>expression</i> {and   or} [not] <i>expression</i> where [not] <i>time_period1</i> overlaps <i>time_period2</i></pre>
while	while <i>logical_expression</i> [plan " <i>abstract plan</i> "] <i>statement</i>
writetext	writetext [[ <i>database.</i> ]owner.] <i>table_name.column_name</i> <i>text_pointer</i> [readpast] [with log] <i>data</i>

## System procedures

See *Reference Manual: Procedures* for details.

sp_activeroles	sp_activeroles [expand_down]
sp_addalias	sp_addalias <i>loginame, name_in_db</i>

sp_addauditrecord	sp_addauditrecord [ <i>text</i> [, <i>db_name</i> [, <i>obj_name</i> [, <i>owner_name</i> [, <i>dbid</i> [, <i>objid</i> ]]]]]]
sp_addaudittable	sp_addaudittable <i>devname</i>
sp_addengine	sp_addengine <i>engine_number</i> , <i>engine_group</i>
sp_addexeclass	sp_addexeclass <i>classname</i> , <i>priority</i> , <i>timeslice</i> , <i>engine_group</i>
sp_addextendedproc	sp_addextendedproc <i>esp_name</i> , <i>dll_name</i>
sp_addexternlogin	sp_addexternlogin <i>remote_server</i> , <i>login_name</i> , <i>remote_name</i> [, <i>remote_password</i> ] [ <i>role_name</i> ]
sp_addgroup	sp_addgroup <i>grpname</i>
sp_addlanguage	sp_addlanguage <i>language</i> , <i>alias</i> , <i>months</i> , <i>shortmons</i> , <i>days</i> , <i>datefmt</i> , <i>datefirst</i>
sp_addlogin	sp_addlogin <i>loginame</i> , <i>passwd</i> [, <i>defdb</i> ] [, <i>deflanguage</i> ] [, <i>fullname</i> ] [, <i>passwdexp</i> ] [, <i>minpwrlen</i> ] [, <i>maxfailedlogins</i> ]
sp_addmessage	sp_addmessage <i>message_num</i> , <i>message_text</i> [, <i>language</i> [, <i>with_log</i> [, <i>replace</i> ]]]
sp_addobjectdef	sp_addobjectdef <i>tablename</i> , "objectdef" [, "objecttype"]
sp_add_qpgroup	sp_add_qpgroup <i>new_name</i>
sp_addremotelogin	sp_addremotelogin <i>remoteserver</i> [, <i>loginame</i> [, <i>remotename</i> ]]
sp_add_resource_limit	sp_add_resource_limit <i>name</i> , <i>appname</i> , <i>rangename</i> , <i>limittype</i> , <i>limitvalue</i> [, <i>enforced</i> [, <i>action</i> [, <i>scope</i> ]]]
sp_addsegment	sp_addsegment <i>segname</i> , <i>dbname</i> , <i>devname</i>
sp_addserver	sp_addserver " <i>lname</i> " [, <i>class</i> [, " <i>pname</i> "]]
sp_addthreshold	sp_addthreshold <i>dbname</i> , <i>segname</i> , <i>free_space</i> , <i>proc_name</i>
sp_add_time_range	sp_add_time_range <i>name</i> , <i>startday</i> , <i>endday</i> , <i>starttime</i> , <i>endtime</i>
sp_addtype	sp_addtype <i>typename</i> , <i>phystype</i> [( <i>length</i> )   ( <i>precision</i> [, <i>scale</i> ])] [, "identity"   <i>nulltype</i> ]
sp_addumpdevice	sp_addumpdevice {"tape"   "disk"}, <i>logicalname</i> , <i>physicalname</i> [, <i>tapesize</i> ]
sp_adduser	sp_adduser <i>loginame</i> [, <i>name_in_db</i> [, <i>grpname</i> ]]
sp_altermessage	sp_altermessage <i>message_id</i> , <i>parameter</i> , <i>parameter_value</i>
sp_audit	sp_audit <i>option</i> , <i>login_name</i> , <i>object_name</i> [, <i>setting</i> ]
sp_autoconnect	sp_autoconnect <i>server</i> , {true false} [, <i>loginame</i> ]
sp_bindcache	sp_bindcache <i>cachename</i> , <i>dbname</i> [, [ <i>ownername</i> .] <i>tablename</i> ] [, <i>indexname</i>   "text only"]]

---

sp_bindefault	sp_bindefault <i>defname, objname</i> [, futureonly]
sp_bindexeclass	sp_bindexeclass " <i>object_name</i> ", " <i>object_type</i> ", " <i>scope</i> ", " <i>classname</i> "
sp_bindmsg	sp_bindmsg <i>constrname, msgid</i>
sp_bindrule	sp_bindrule <i>rulename, objname</i> [, futureonly]
sp_cacheconfig	sp_cacheconfig [ <i>cachename</i> [, " <i>cache_size</i> {P K M G}"] [, logonly   mixed ] [, strict   relaxed ] ] [, " <i>cache_partition</i> ={1 2 4 8 16 32 64}"]
sp_cachestrategy	sp_cachestrategy <i>dbname</i> , [ <i>ownername</i> .] <i>tablename</i> [, <i>indexname</i>   "text only"   "table only" [, { prefetch   mru }, { "on"   "off"}]]
sp_changedbowner	sp_changedbowner <i>loginame</i> [, true ]
sp_changegroup	sp_changegroup <i>grpname, username</i>
sp_checknames	sp_checknames
sp_checkreswords	sp_checkreswords [ <i>user_name_param</i> ]
sp_checksourc	sp_checksourc [ <i>objname</i> [, <i>tabname</i> [, <i>username</i> ]]]
sp_chgattribute	sp_chgattribute <i>objname</i> , {"max_rows_per_page"   "fillfactor"   "reserverpagegap"   "exp_row_size" concurrency_opt_threshold   "optimistic_index_lock"}, <i>value</i> , <i>optvalue</i>  sp_chgattribute " <i>table_name</i> ", "identity_gap", <i>set_number</i> sp_chgattribute < <i>table_name</i> >, "dealloc_first_txtpg",1
sp_clearpsex	sp_clearpsex <i>spid, exeattr</i>
sp_clearstats	sp_clearstats [ <i>loginame</i> ]
sp_client_addr	sp_client addr[" <i>spid</i> "]
sp_cmp_all_qplans	sp_cmp_all_qplans <i>group1, group2</i> [, <i>mode</i> ]
sp_cmp_qplans	sp_cmp_qplans <i>id1, id2</i>
sp_commonkey	sp_commonkey <i>tabaname, tabbname, col1a, col1b</i> [, <i>col2a, col2b, ..., col8a, col8b</i> ]
sp_companion	sp_companion [ <i>server_name</i> {, configure [, {with_proxydb   NULL}] [, <i>srvlogin</i> ] [, <i>server_password</i> ] [, <i>cluster_login</i> ] [, <i>cluspassword</i> ]]   drop   suspend   resume

	prepare_failback   do_advisory} {, all   help   group attribute_name   base attribute_name}
sp_configure	sp_configure [ <i>configname</i> [, <i>configvalue</i> ]   <i>group_name</i>   <i>non_unique_parameter_fragment</i> ][number of histogram steps, <i>n</i> ] sp_configure "configuration file", 0, {"write"   "read"   "verify"   "restore"} " <i>file_name</i> "
sp_copy_all_qplans	sp_copy_all_qplans <i>src_group</i> , <i>dest_group</i>
sp_copy_qplan	sp_copy_qplan <i>src_id</i> , <i>dest_group</i>
sp_countmetadata	sp_countmetadata " <i>configname</i> " [, <i>dbname</i> ]
sp_cursorinfo	sp_cursorinfo [{ <i>cursor_level</i>   null}] [, <i>cursor_name</i> ]
sp_dbextend	sp_dbextend ['check'   'clear'   'disable'   'enable'   'execute'   'help'   'list'   'listfull'   'modify'   'reload defaults'   'set'   'simulate'   'trace'   'version'   'who']
sp_dboption	sp_dboption [ <i>dbname</i> , <i>optname</i> , {true   false}]
sp_dbrecovery_order	sp_dbrecovery_order [ <i>database_name</i> [, <i>rec_order</i> [, force]]]
sp_dbremap	sp_dbremap <i>dbname</i>
sp_defaultloc	sp_defaultloc <i>dbname</i> , {"defaultloc"   NULL} [, " <i>defaulttype</i> "]
sp_depends	sp_depends <i>objname</i> [, <i>column_name</i> ]
sp_deviceattr	sp_deviceattr <i>logicalname</i> , <i>optname</i> , <i>optvalue</i>
sp_diskdefault	sp_diskdefault <i>logicalname</i> , {defaulton   defaultoff}
sp_displayaudit	sp_displayaudit ["procedure"   "object"   "login"   "database"   "global"   "default_object"   "default_procedure" [, " <i>name</i> "]]
sp_displaylevel	sp_displaylevel [ <i>loginame</i> [, <i>level</i> ]]
sp_displaylogin	sp_displaylogin [ <i>loginame</i> [, expand_up   expand_down]]
sp_displayroles	sp_displayroles [ <i>grantee_name</i> [, <i>mode</i> ]]
sp_dropalias	sp_dropalias <i>loginame</i>
sp_drop_all_qplans	sp_drop_all_qplans <i>name</i>
sp_dropdevice	sp_dropdevice <i>logicalname</i>
sp_dropengine	sp_dropengine <i>engine_number</i> , <i>engine_group</i>
sp_dropexeclss	sp_dropexeclss <i>classname</i>
sp_dropextendedproc	sp_dropextendedproc <i>esp_name</i>
sp_dropexternlogin	sp_dropexternlogin <i>remote_server</i> [, <i>login_name</i> ] [, <i>role_name</i> ]

---

sp_dropglockpromote	sp_dropglockpromote {"database"   "table"}, <i>objname</i>
sp_dropgroup	sp_dropgroup <i>grpname</i>
sp_dropkey	sp_dropkey <i>keytype</i> , <i>tablename</i> [, <i>deptabname</i> ]
sp_droplanguage	sp_droplanguage <i>language</i> [, <i>dropmessages</i> ]
sp_droplogin	sp_droplogin <i>loginame</i>
sp_dropmessage	sp_dropmessage <i>message_num</i> [, <i>language</i> ]
sp_dropobjectdef	sp_dropobjectdef " <i>object_name</i> "
sp_drop_qpgroup	sp_drop_qpgroup <i>group</i>
sp_drop_qplan	sp_drop_qplan <i>id</i>
sp_dropremotelogin	sp_dropremotelogin <i>remoteserver</i> [, <i>loginame</i> [, <i>remotename</i> ]]
sp_drop_resource_limit	sp_drop_resource_limit { <i>name</i> , <i>appname</i> } [, <i>rangename</i> , <i>limittype</i> , <i>enforced</i> , <i>action</i> , <i>scope</i> ]
sp_droprowlockpromote	sp_droprowlockpromote {"database"   "table"}, <i>objname</i>
sp_dropsegment	sp_dropsegment <i>segname</i> , <i>dbname</i> [, <i>device</i> ]
sp_dropserver	sp_dropserver <i>server</i> [, <i>droplogins</i> ]
sp_droptreshold	sp_droptreshold <i>dbname</i> , <i>segname</i> , <i>free_space</i>
sp_drop_time_range	sp_drop_time_range <i>name</i>
sp_droptype	sp_droptype <i>typename</i>
sp_dropuser	sp_dropuser <i>name_in_db</i>
sp_dumpoptimize	sp_dumpoptimize [ 'archive_space = {maximum   minimum   default }' ] sp_dumpoptimize [ 'reserved_threshold = { <i>nnn</i>   default }' ] sp_dumpoptimize [ 'allocation_threshold = { <i>nnn</i>   default }' ]
sp_engine	sp_engine {"online"   [offline   can_offline] [, <i>engine_id</i> ]   ["shutdown", <i>engine_id</i> ]}
sp_estspace	sp_estspace <i>table_name</i> , <i>no_of_rows</i> , <i>fill_factor</i> , <i>cols_to_max</i> , <i>textbin_len</i> , <i>iosec</i> , <i>page_size</i>
sp_export_qpgroup	sp_export_qpgroup <i>usr</i> , <i>group</i> , <i>tab</i>
sp_extendsegment	sp_extendsegment <i>segname</i> , <i>dbname</i> , <i>devname</i>
sp_extengine	sp_extengine ' <i>ejb_server</i> ', '{ start   stop   status }'
sp_familylock	sp_familylock [ <i>fpid1</i> [, <i>fpid2</i> ]]
sp_find_qplan	sp_find_qplan <i>pattern</i> [, <i>group</i> ]
sp_fixindex	sp_fixindex <i>dbname</i> , <i>table_name</i> , <i>index_id</i>
sp_flushstats	sp_flushstats <i>objname</i>

sp_forceonline_db	sp_forceonline_db <i>dbname</i> , {"sa_on"   "sa_off"   "all_users"}
sp_forceonline_object	sp_forceonline_object <i>dbname</i> , <i>objname</i> , <i>indid</i> , {sa_on   sa_off   all_users} [, no_print]
sp_forceonline_page	sp_forceonline_page <i>dbname</i> , <i>pgid</i> , {"sa_on"   "sa_off"   "all_users"}
sp_foreignkey	sp_foreignkey <i>tablename</i> , <i>pktablename</i> , <i>col1</i> [, <i>col2</i> ] ... [, <i>col8</i> ]
sp_freedll	sp_freedll <i>dll_name</i>
sp_getmessage	sp_getmessage <i>message_num</i> , <i>result</i> output [, <i>language</i> ]
sp_grantlogin	sp_grantlogin { <i>login_name</i>   <i>group_name</i> } ["role_list"   default]
sp_ha_admin	sp_ha_admin [cleansessions   help]
sp_help	sp_help [ <i>objname</i> ]
sp_helppartition	sp_helppartition [ <i>table_name</i> ]
sp_helpcache	sp_helpcache { <i>cache_name</i>   "cache_size[P K M G]"}
sp_helpconfig	sp_helpconfig " <i>configname</i> ", ["size"]
sp_helpconstraint	sp_helpconstraint [ <i>objname</i> ] [, detail]
sp_helpdb	sp_helpdb [ <i>dbname</i> ]
sp_helpdevice	sp_helpdevice [ <i>devname</i> ]
sp_helpextendedproc	sp_helpextendedproc [ <i>esp_name</i> ]
sp_helpexternlogin	sp_helpexternlogin [ <i>remote_server</i> ] [, <i>login_name</i> ] [, <i>role_name</i> ]
sp_helpgroup	sp_helpgroup [ <i>grpname</i> ]
sp_helpindex	sp_helpindex <i>objname</i>
sp_helpjava	sp_helpjava ["class" [, <i>java_class_name</i> [, "detail"   "depends" ] ]   "jar" [, <i>jar_name</i> [, "depends" ] ] ]
sp_helpjoins	sp_helpjoins <i>lefttab</i> , <i>righttab</i>
sp_helpkey	sp_helpkey [ <i>tablename</i> ]
sp_helplanguage	sp_helplanguage [ <i>language</i> ]
sp_helplog	sp_helplog
sp_helpobjectdef	sp_helpobjectdef [ <i>object_name</i> ]
sp_help_qpgroup	sp_help_qpgroup [ <i>group</i> [, <i>mode</i> ] ]
sp_help_qplan	sp_help_qplan <i>id</i> [, <i>mode</i> ]
sp_helpremotelogin	sp_helpremotelogin [ <i>remoteserver</i> ] [, <i>remotename</i> ]
sp_help_resource_limit	sp_help_resource_limit [ <i>name</i> [, <i>appname</i> [, <i>limittime</i> [, <i>limitday</i> [, <i>scope</i> [, <i>action</i> ]]]]]]



---

sp_helprotect	sp_helprotect [ <i>name</i> [, <i>username</i> [, "grant" [, "none" "granted" "enabled" "role_name"]]]]
sp_helpsegment	sp_helpsegment [ <i>segname</i> ]
sp_helpserver	sp_helpserver [ <i>server</i> ]
sp_helpsort	sp_helpsort
sp_helptext	sp_helptext <i>objname</i> [, <i>number</i> ]
sp_helpthreshold	sp_helpthreshold [ <i>segname</i> ]
sp_helpuser	sp_helpuser [ <i>name_in_db</i> ]
sp_hidetext	sp_hidetext [ <i>objname</i> [, <i>tablename</i> [, <i>username</i> ]]]
sp_import_qpgroup	sp_import_qpgroup <i>tab</i> , <i>usr</i> , <i>group</i>
sp_indsuspect	sp_indsuspect [ <i>tab_name</i> ]
sp_ldapadmin	sp_ldapadmin { set_primary_url, ' <i>ldapurl</i>   set_secondary_url, { ' <i>ldapurl</i> '   null }   list_urls   check_url, ' <i>ldapurl</i>   check_login, ' <i>login_name</i> ' }
	<i>ldapurl</i> ::= <i>ldap://host:port/node/?attributes?base</i>   <i>one</i>   <i>sub?filter</i>
sp_listener	sp_listener " <i>command</i> ", " <i>server_name</i> ", <i>engine</i>   <i>remaining</i>
	Or:
	sp_listener " <i>command</i> ", "[ <i>protocol</i> :] <i>machine:port</i> ", <i>engine</i>
sp_listsuspect_db	sp_listsuspect_db
sp_listsuspect_object	sp_listsuspect_object [ <i>dbname</i> ]
sp_listsuspect_page	sp_listsuspect_page [ <i>dbname</i> ]
sp_lock	sp_lock [ <i>spid1</i> [, <i>spid2</i> ]]
sp_locklogin	sp_locklogin [ <i>loginame</i> , "{lock   unlock}"]
sp_logdevice	sp_logdevice <i>dbname</i> , <i>devname</i>
sp_loginconfig	sp_loginconfig [" <i>parameter_name</i> "]
sp_logininfo	sp_logininfo [" <i>login_name</i> "   " <i>group_name</i> "]
sp_logiosize	sp_logiosize ["default"   " <i>size</i> "   "all"]
sp_modifylogin	sp_modifylogin { <i>loginame</i>   "all overrides"}, <i>option</i> , <i>value</i>
sp_modify_resource_limit	sp_modify_resource_limit { <i>name</i> , <i>appname</i> } <i>rangename</i> , <i>limittype</i> , <i>limitvalue</i> , <i>enforced</i> , <i>action</i> , <i>scope</i>
sp_modify_time_range	sp_modify_time_range <i>name</i> , <i>startday</i> , <i>endday</i> , <i>starttime</i> , <i>endtime</i>

sp_modifystats	<p>sp_modifystats [database].[owner].table_name,          {"column_group"   "all"},          MODIFY_DENSITY,          {range   total},          {absolute   factor},          "value"</p> <p>sp_modifystats [database].[owner].table_name,          column_name,          REMOVE_SKEW_FROM_DENSITY</p>
sp_modifythreshold	sp_modifythreshold dbname, segname, free_space [, new_proc_name] [, new_free_space] [, new_segname]
sp_monitor	sp_monitor
sp_monitorconfig	sp_monitorconfig "configname" [ , "result_tbl_name"]
sp_object_stats	sp_object_stats interval [, top_n [, dbname, objname [, rpt_option ]]]
sp_passthru	sp_passthru server, command, errcode, errmsg, rowcount [, arg1, arg2, ... argn]
sp_password	sp_password caller_passwd, new_passwd [, loginame]
sp_placeobject	sp_placeobject segname, objname
sp_plan_dbccdb	sp_plan_dbccdb [dbname]
sp_poolconfig	<p>To create a memory pool in an existing cache, or to change pool size:</p> <p>sp_poolconfig cache_name [, "mem_size [P K M G]", "config_poolK"          [, "affected_poolK"]]</p> <p>To change a pool's wash size:</p> <p>sp_poolconfig cache_name, "io_size ", "wash=size[P K M G]"</p> <p>To change a pool's asynchronous prefetch percentage:</p> <p>sp_poolconfig cache_name, "io_size ",          "local async prefetch limit=percent "</p>
sp_primarykey	sp_primarykey tablename, col1 [, col2, col3, ..., col8]
sp_processmail	sp_processmail [subject] [, originator [, dbuser [, dbname [, filetype [, separator]]]]]
sp_procxmode	sp_procxmode [procname [, tranmode]]
sp_recompile	sp_recompile objname
sp_remap	sp_remap objname
sp_remoteoption	sp_remoteoption [remoteserver [, loginame [, remotename [, optname [, optvalue]]]]]
sp_remotesql	sp_remotesql server, query [, query2, ... , query254]
sp_rename	sp_rename objname, newname [, "index"   "column"]

---

sp_renamedb	sp_renamedb <i>dbname, newname</i>
sp_rename_qpgroup	sp_rename_qpgroup <i>old_name, new_name</i>
sp_reportstats	sp_reportstats [ <i>loginame</i> ]
sp_revokelgin	sp_revokelgin { <i>login_name</i>   <i>group_name</i> }
sp_role	sp_role {"grant"   "revoke"}, <i>rolename, loginame</i>
sp_sendmsg	sp_sendmsg <i>ip_address, port_number, message</i>
sp_serveroption	sp_serveroption [ <i>server, optname, optvalue</i> ]
sp_setlangalias	sp_setlangalias <i>language, alias</i>
sp_setpglockpromote	sp_setpglockpromote {"database"   "table"}, <i>objname, new_lwm, new_hwm, new_pct</i> sp_setpglockpromote server, NULL, <i>new_lwm, new_hwm, new_pct</i>
sp_setpsex	sp_setpsex <i>spid, exeattr, value</i>
sp_set_qplan	sp_set_qplan <i>id, plan</i>
sp_setrowlockpromote	sp_setrowlockpromote "server", NULL, <i>new_lwm, new_hwm, new_pct</i> sp_setrowlockpromote {"database"   "table"}, <i>objname, new_lwm, new_hwm, new_pct</i>
sp_setsuspect_granularity	sp_setsuspect_granularity [ <i>dbname</i> [, "database"   "page" [, "read_only"]]]
sp_setsuspect_threshold	sp_setsuspect_threshold [ <i>dbname</i> [, <i>threshold</i> ]]
sp_showcontrolinfo	sp_showcontrolinfo [ <i>object_type, object_name, spid</i> ]
sp_showexeclass	sp_showexeclass [ <i>execlassname</i> ]
sp_showplan	sp_showplan <i>spid, batch_id</i> output, <i>context_id</i> output, <i>stmt_num</i> output
sp_showpsex	sp_showpsex [ <i>spid</i> ]
sp_spaceused	sp_spaceused [ <i>objname</i> [,1] ]
sp_ssladmin	sp_ssladmin [addcert, <i>certificate_path</i> [, <i>password</i>   NULL]] sp_ssladmin [dropcert, <i>certificate_path</i> ] sp_ssladmin [lscert] sp_ssladmin [help]
sp_syntax	sp_syntax <i>word</i> [, <i>mod</i> ][, <i>language</i> ]
sp_sysmon	sp_sysmon begin_sample sp_sysmon { <i>end_sample</i>   <i>interval</i> } [, <i>section</i> [, <i>applmon</i> ] ] sp_sysmon { <i>end_sample</i>   <i>interval</i> } [, <i>applmon</i> ]

sp_tempdb	<pre> sp_tempdb [   [ { create   drop } , <i>groupname</i> ]     [ { add   remove } , <i>tempdbname</i>, <i>groupname</i> ]     [ { bind, <i>objtype</i>, <i>objname</i>, <i>bindtype</i>, <i>bindobj</i> [, <i>scope</i>, <i>hardness</i> ] }       { unbind, <i>objtype</i>, <i>objname</i> [, <i>scope</i> ] } ]     [ unbindall_db, <i>tempdbname</i> ]     [ show [, "all"   "gr"   "db"   "login"   "app" [, <i>name</i> ] ]     [ who, <i>dbname</i> ]   [ help ] ] </pre>
sp_thresholdaction	<pre> sp_thresholdaction @<i>dbname</i>,   @<i>segment_name</i>,   @<i>space_left</i>,   @<i>status</i> </pre>
sp_transactions	<pre> sp_transactions ["xid", <i>xid_value</i>]     ["state", {"heuristic_commit"   "heuristic_abort"     "prepared"   "indoubt"} [, "xactname"]]     ["gtrid", <i>gtrid_value</i>] </pre>
sp_unbindcache	<pre> sp_unbindcache <i>dbname</i> [, [<i>owner</i>].<i>tablename</i>   [, <i>indexname</i>   "text only"]] </pre>
sp_unbindcache_all	<pre> sp_unbindcache_all <i>cache_name</i> </pre>
sp_unbindefault	<pre> sp_unbindefault <i>objname</i> [, futureonly] </pre>
sp_unbindexclass	<pre> sp_unbindexclass <i>object_name</i>, <i>object_type</i>, <i>scope</i> </pre>
sp_unbindmsg	<pre> sp_unbindmsg <i>constrname</i> </pre>
sp_unbindrule	<pre> sp_unbindrule <i>objname</i> [, futureonly [, "accessrule"   "all"]] </pre>
sp_volchanged	<pre> sp_volchanged <i>session_id</i>, <i>devname</i>, <i>action</i> [, <i>fname</i> [, <i>vname</i>]] </pre>
sp_who	<pre> sp_who [<i>loginame</i>   "<i>spid</i>"] </pre>

## Catalog stored procedures

See *Reference Manual: Procedures* for details.

sp_column_privileges	<pre> sp_column_privileges <i>table_name</i> [, <i>table_owner</i>   [, <i>table_qualifier</i> [, <i>column_name</i>]]] </pre>
sp_columns	<pre> sp_columns <i>table_name</i> [, <i>table_owner</i> ]   [, <i>table_qualifier</i> [, <i>column_name</i>] </pre>
sp_databases	<pre> sp_databases </pre>
sp_datatype_info	<pre> sp_datatype_info [<i>data_type</i>] </pre>

---

sp_fkeys	sp_fkeys <i>pktable_name</i> [, <i>pktable_owner</i> ] [, <i>pktable_qualifier</i> ] [, <i>fktable_name</i> ] [, <i>fktable_owner</i> ] [, <i>fktable_qualifier</i> ]
sp_pkeys	sp_pkeys <i>table_name</i> [, <i>table_owner</i> ] [, <i>table_qualifier</i> ]
sp_server_info	sp_server_info [ <i>attribute_id</i> ]
sp_special_columns	sp_special_columns <i>table_name</i> [, <i>table_owner</i> ] [, <i>table_qualifier</i> ] [, <i>col_type</i> ]
sp_sproc_columns	sp_sproc_columns <i>procedure_name</i> [, <i>procedure_owner</i> ] [, <i>procedure_qualifier</i> ] [, <i>column_name</i> ]
sp_statistics	sp_statistics <i>table_name</i> [, <i>table_owner</i> ] [, <i>table_qualifier</i> ] [, <i>index_name</i> ] [, <i>is_unique</i> ]
sp_stored_procedures	sp_stored_procedures [ <i>sp_name</i> [, <i>sp_owner</i> [, <i>sp_qualifier</i> ]]]
sp_table_privileges	sp_table_privileges <i>table_name</i> [, <i>table_owner</i> ] [, <i>table_qualifier</i> ]
sp_tables	sp_tables [ <i>table_name</i> ] [, <i>table_owner</i> ] [, <i>table_qualifier</i> ] [, <i>table_type</i> ]

## System Extended Stored Procedures

See *Reference Manual: Procedures* for details.

xp_cmdshell	xp_cmdshell <i>command</i> [, <i>no_output</i> ]
xp_deletemail	xp_deletemail [ <i>msg_id</i> ]
xp_enumgroups	xp_enumgroups [ <i>domain_name</i> ]
xp_findnextmsg	xp_findnextmsg @ <i>msg_id</i> = @ <i>msg_id</i> output [, <i>type</i> ] [, <i>unread_only</i> = {true   false}]
xp_logevent	xp_logevent <i>error_number</i> , <i>message</i> [, <i>type</i> ]
xp_readmail	xp_readmail [ <i>msg_id</i> ] [, <i>recipients</i> output] [, <i>sender</i> output] [, <i>date_received</i> output] [, <i>subject</i> output] [, <i>cc</i> output] [, <i>message</i> output] [, <i>attachments</i> output] [, <i>suppress_attach</i> = {true   false}] [, <i>peek</i> = {true   false}] [, <i>unread</i> = {true   false}] [, <i>msg_length</i> output] [, <i>bytes_to_skip</i> [output]] [, <i>type</i> [output]]

xp_sendmail	xp_sendmail <i>recipient</i> [, <i>recipient</i> ] . . . [, <i>subject</i> ] [, <i>cc_recipient</i> ] . . . [, <i>bcc_recipient</i> ] . . . [, { <i>query</i>   <i>message</i> }] [, <i>attachname</i> ] [, <i>attach_result</i> = {true   false}] [, <i>echo_error</i> = {true   false}] [, <i>include_file</i> [, <i>include_file</i> ] . . .] [, <i>no_column_header</i> = {true   false}] [, <i>no_output</i> = {true   false}] [, <i>width</i> ] [, <i>separator</i> ] [, <i>dbuser</i> ] [, <i>dbname</i> ] [, <i>type</i> ] [, <i>include_query</i> = {true   false}]
xp_startmail	xp_startmail [ <i>mail_user</i> ] [, <i>mail_password</i> ]
xp_stopmail	xp_stopmail

## dbcc stored procedures

See *Reference Manual: Procedures* for details.

sp_dbcc_alterws	sp_dbcc_alterws <i>dbname</i> , <i>wsname</i> , "wssize[K M]"
sp_dbcc_configreport	sp_dbcc_configreport [ <i>dbname</i> ]
sp_dbcc_createws	sp_dbcc_createws <i>dbname</i> , <i>segname</i> , [ <i>wsname</i> ], <i>wstype</i> , "wssize[K M]"
sp_dbcc_deletedb	sp_dbcc_deletedb [ <i>dbname</i>   <i>dbid</i> ]
sp_dbcc_deletehistory	sp_dbcc_deletehistory [ <i>cutoffdate</i> ] [, <i>dbname</i>   <i>dbid</i> ]
sp_dbcc_differentialreport	sp_dbcc_differentialreport [ <i>dbname</i> ] [, <i>objectname</i> ], [ <i>db_op</i> ] [, "date1" [, "date2"]]
sp_dbcc_evaluatedb	sp_dbcc_evaluatedb [ <i>dbname</i> ]
sp_dbcc_faultreport	sp_dbcc_faultreport [ <i>report_type</i> ] [, <i>dbname</i> ] [, <i>objectname</i> ] [, <i>date</i> ] [, @hard_only]]]
sp_dbcc_fullreport	sp_dbcc_fullreport [ <i>dbname</i> ] [, <i>objectname</i> ] [, <i>date</i> ]]]
sp_dbcc_recommendations	sp_dbcc_recommendations <i>dbname</i> [, <i>date</i> ] [, <i>opid</i> ] [, <i>objectname</i> ]]]
sp_dbcc_runcheck	sp_dbcc_runcheck <i>dbname</i> [, <i>user_proc</i> ]

---

sp_dbcc_statisticsreport	sp_dbcc_statisticsreport [dbname [, objectname [, date]]]
sp_dbcc_summaryreport	sp_dbcc_summaryreport [dbname [, date] [, opname ] ]
sp_dbcc_updateconfig	sp_dbcc_updateconfig dbname, type, "str1" [, "str2"]

## System tables

See *Reference Manual: Tables* for details.

sysalternates	All databases – contains one row for each Adaptive Server user mapped (or aliased) to a user of the current database.
sysattributes	All databases – system attributes define properties of objects such as databases, tables, indexes, users, logins, and procedures.
sysauditoptions	sybsecurity database – contains one row for each server-wide audit option and indicates the current setting for that option.
sysaudits_01 – sysaudits_08	sybsecurity database – contain the audit trail.
syscharsets	master database only – contains one row for each character set and sort order defined for use by Adaptive Server.
syscolumns	All databases – contains one row for every column in every table and view, and a row for each parameter in a procedure.
syscomments	All databases – contains entries for each view, rule, default, trigger, table constraint, and procedure.
sysconfigures	master database only – contains one row for each configuration parameter that can be set by the user.
sysconstraints	All databases – has one row for each referential constraint and check constraint associated with a table or column.
syscoordinations	sybssystemdb database only – contains information about remote Adaptive Servers participating in distributed transactions (remote participants) and their coordination states.
syscurconfigs	master database only – syscurconfigs is built dynamically when queried. It contains an entry for each of the configuration parameters, as does sysconfigures, but with the current values rather than the default values.
sysdatabases	master database only – contains one row for each database in Adaptive Server.

sysdepends	All databases – contains one row for each procedure, view, or table that is referenced by a procedure, view, or trigger.
sysdevices	master database only – contains one row for each tape dump device, disk dump device, disk for databases, and disk partition for databases.
sysengines	master database only – contains one row for each Adaptive Server engine currently online.
sysgams	All databases – stores the global allocation map (GAM) for the database. The GAM stores a bitmap for all allocation units of a database, with one bit per allocation unit. You cannot select from or view sysgams.
sysindexes	All databases – contains one row for each clustered index, one row for each nonclustered index, one row for each table that has no clustered index, and one row for each table that contains text or image columns.
sysjars	All databases – contains one row for each Java archive (JAR) file that is retained in the database. Uses row-level locking.
syskeys	All databases – contains one row for each primary, foreign, or common key.
syslanguages	master database only – contains one row for each language known to Adaptive Server. us_english is not in syslanguages, but it is always available to Adaptive Server.
syslisteners	master database only – contains a row for each network protocol available for connecting with the current Adaptive Server. Adaptive Server builds syslisteners dynamically when a user or client application queries the table.
syslocks	master database only – contains information about active locks. It is built dynamically when queried by a user. No updates to syslocks are allowed.
sysloginroles	master database only – contains a row for each instance of a server login possessing a system role.
syslogins	master database only – contains one row for each valid Adaptive Server user account.
syslogs	All databases – contains the transaction log. It is used by Adaptive Server for recovery and roll forward. It is not useful to users.
syslogshold	master database only – contains information about each database's oldest active transaction (if any) and the Replication Server truncation point (if any) for the transaction log, but it is not a normal table. Rather, it is built dynamically when queried by a user. No updates to syslogshold are allowed.
sysmessages	master database only – contains one row for each system error or warning that can be returned by Adaptive Server.



---

sysmonitors	master database only – contains one row for each monitor counter.
sysobjects	All databases – contains one row for each table, view, stored procedure, extended stored procedure, log, rule, default, trigger, check constraint, referential constraint, and (in tempdb only) temporary object.
syspartitions	All databases – contains one row for each partition (page chain) of a partitioned table.
sysprocedures	All databases – contains entries for each view, default, rule, trigger, procedure, declarative default, and check constraint.
sysprocesses	master database only – contains information about Adaptive Server processes, but it is not a normal table. It is built dynamically when queried by a user.
sysprotects	All databases – contains information on permissions that have been granted to, or revoked from, users, groups, and roles.
sysqueryplans	All databases – contains two or more rows for each abstract query plan. Uses datarow locking.
sysreferences	All databases – contains one row for each referential integrity constraint declared on a table or column.
sysremotelogins	master database only – contains one row for each remote user that is allowed to execute remote procedure calls on this Adaptive Server.
sysresourcelimits	master database only – contains a row for each resource limit defined by Adaptive Server.
sysroles	All databases – maps server role IDs to local role IDs.
syssecmechs	master database only – contains information about the security services supported by each security mechanism that is available to Adaptive Server.
syssegments	All databases – contains one row for each segment (named collection of disk pieces).
syssservers	master database only – contains one row for each remote Adaptive Server, Backup Server™, or Open Server™ on which this Adaptive Server can execute remote procedure calls.
syssessions	master database only – is only used when Adaptive Server is configured for Sybase's Failover in a high availability system.
sysssrvroles	master database only – contains a row for each system or user-defined role.
sysstatistics	All databases – contains one or more rows for each indexed column on a user table.

sysabstats	All databases – contains one row for each clustered index, one row for each nonclustered index, and one row for each table that has no clustered index.
systhresholds	All databases – contains one row for each threshold defined for the database.
sys timeranges	master database only – stores named time ranges, which are used by Adaptive Server to control when a resource limit is active.
sys transactions	master database only – contains information about Adaptive Server transactions, but it is not a normal table.
sys types	All databases – contains one row for each system-supplied and user-defined datatype.
sys usages	master database only – contains one row for each disk allocation piece assigned to a database.
sys usermessages	All databases – contains one row for each user-defined message that can be returned by Adaptive Server.
sys users	All databases – contains one row for each user allowed in the database, and one row for each group or role.
sys xtypes	All databases – contains one row for each extended, Java-SQL datatype. Uses row-level locking.
sys licenseslog	master database only – contains one row for each update of the maximum number of licenses used in Adaptive Server per 24-hour period.

## **dbcc tables**

See *Reference Manual: Tables* for details.

dbcc_config	The dbcc_config table describes the currently executing or last completed dbcc checkstorage operation. It defines: <ul style="list-style-type: none"><li>• The location of resources dedicated to the dbcc checkstorage operation</li><li>• Resource usage limits for the dbcc checkstorage operation</li></ul>
dbcc_counters	The dbcc_counters table stores the results of the analysis performed by dbcc checkstorage. Counters are maintained for each database, table, index, partition, device, and invocation of dbcc.
dbcc_fault_params	The dbcc_fault_params table provides additional descriptive information for a fault entered in the dbcc_faults table.

---

dbcc_faults	The dbcc_faults table provides a description of each fault detected by dbcc checkstorage.
dbcc_operation_log	The dbcc_operation_log table records the use of the dbcc checkstorage operations.
dbcc_operation_results	The dbcc_operation_results table provides additional descriptive information for an operation recorded in the dbcc_operation_log table.
dbcc_types	The dbcc_types table provides the definitions of the data types used by dbcc checkstorage. This table is not actually used by the dbcc stored procedures. It is provided to facilitate the use of the other tables in dbccdb, and to document the semantics of the data types. Type codes for operation configuration, analysis data reported, fault classification, and fault report parameters are included. If you create your own stored procedures for generating reports, the values listed in the type_name column can be used as report headings.

## Utilities

See *Utility Guide* for details.

backupserver      bcksrvr.exe in Windows NT

```

backupserver
  [-C server_connections]
  [-S b_servername]
  [-I interfaces_file]
  [-e error_log_file]
  [-M sybmultbuf_binary]
  [-N network_connections]
  [-T trace_value]
  [-L Sybase_language_name]
  [-J Sybase_character_set_name]
  [-c tape_config_file]
  [-D n]
  [-A pathname]
  [-P active_service_threads]
  [-V level_number]
  [-p n]
  [-m max_shared_memory]

```

Or

```
backupserver -v
```

bcp      Also use for bcp\_r, bcp\_dce

```
bcp [[database_name.]owner.]table_name [:slice_number] {in | out}
datafile
[-m maxerrors]
[-f formatfile]
[-e errfile]
[-F firstrow]
[-L lastrow]
[-b batchsize]
[-n]
[-c]
[-t field_terminator]
[-r row_terminator]
-U username
[-P password]
[-I interfaces_file]
[-S server]
[-a display_charsel]
[-z language]
[-A packet_size]
[-J client_charset]
[-T text_or_image_size]
[-E]
[-g id_start_value]
[-N]
[-X]
[-K keytab_file]
[-R remote_server_principal]
[-V [security_options]]
[-Z security_mechanism]
[-Q]
[-Y]
```

Or

```
bcp -v
```

---

**Note** [-N] is for Windows NT only.

---

buildmaster

See dataserver for UNIX, or sqlsrvr for Windows NT.

certauth

```
certauth
[-r]
[-C caCert_file]
[-Q request_filename]
[-K caKey_filename]
[-O SignedCert_filename]
[-P caPassword]
[-T valid_time]
```

Or

---

certauth -v  
certpk12  
    {-O *Pkcs12\_file* | -I *Pkcs12\_file*}  
    [-C *Cert\_file*]  
    [-K *Key\_file*]  
    [-P *key\_password*]  
    [-E *Pkcs12\_password*]

Or

certpk12 -v  
certreq  
    [-F *input\_file*]  
    [-R *request\_filename*]  
    [-K *PK\_filename*]  
    [-P *password*]

Or

certreq -v  
charset  
    UNIX platforms only.

charset  
    [-P *password*]  
    [-S *server*]  
    [-l *interface*]  
    *sort\_order*  
    [ *charset* ]

Or

charset -v  
cobpre  
    Also use for cobpre\_r and cobpre\_dce.

cobpre [-a] [-b] [-c] [-d] [-e]  
    [-f] [-h] [-m] [-p] [-q] [-r]  
    [-v] [-w] [-x]  
    [-B *compatibility\_mode*]  
    [-C *compiler*]  
    [-D *database*]  
    [-F *fips\_level*]  
    [-G [*isql\_file\_name*]]  
    [-I *include\_path\_name*]  
    [-J *charset\_locale\_name*]  
    [-K *syntax\_level*]  
    [-L [*listing\_file\_name*]]  
    [-N *interfacefile\_name*]  
    [-O *target\_file\_name*]  
    [-P *password*]  
    [-S *server\_name*]  
    [-T *tag*]

```
[ -U userid ]  
[ -V version_number ]  
[ -X application_name ]  
[ -Z language_locale_name ]  
[ @options_file ]  
filename[.ext] ...
```

cpre Also use for cpre\_r and cpre\_dce.

```
cpre [ -a ] [ -b ] [ -c ] [ -d ] [ -f ] [ -h ]  
[ -l ] [ -m ] [ -p ] [ -r ] [ -s ] [ -v ]  
[ -w ] [ -x ]  
[ -B compatibility_mode ]  
[ -C compiler ]  
[ -D database ]  
[ -F fips_level ]  
[ -G [isql_file_name] ]  
[ -I include_path_name ]  
[ -J charset_locale_name ]  
[ -K syntax_level ]  
[ -L [listing_file_name] ]  
[ -N interfacefile_name ]  
[ -O target_file_name ]  
[ -P password ]  
[ -S server_name ]  
[ -T tag ]  
[ -U userid ]  
[ -V version_number ]  
[ -X application_name ]  
[ -Z language_locale_name ]  
[ @options_file ]  
filename[.ext]
```

dataserver UNIX platforms only.

```
dataserver [-f] [-g] [-G] [-h] [-H] [-m] [-q] [-v] [-X]  
[ -a path_to_CAPs_directive_file ]  
[ -b master_device_size [k | K | m | M | g | G] ]  
[ -c config_file_for_server ]  
[ -d device_name ]  
[ -e path_to_error_log ]  
[ -i interfaces_file_directory ]  
[ -K keytab_file ]  
[ -L config_file_name_for_connectivity ]  
[ -M shared_memory_repository_directory ]  
[ -p sa_login_name ]  
[ -r mirror_disk_name ]  
[ -s server_name ]  
[ -T trace_flag ]  
[ -u sa/sso_name ]  
[ -w master | model database ]
```

---

`[-y [password] ]`  
`[-z page_size [ k | K ]]`

Or

`dataserver -v`

ddlgen

`ddlgen`  
`-Ulogin`  
`-Ppassword`  
`-Shost_name : port_number`  
`[-Tobject_type]`

For a list of valid object types, see “ddlgen” in the *Utility Guide*.

`[-Nobject_name]`  
`[-Ddbname]`  
`[-Xextended_object_type]`  
`[-Ooutput_file]`  
`[-Error_file]`  
`[-Lprogress_log_file]`  
`[-Jclient_charset]`  
`-F [% | SGM | GRP | USR | R | D | UDD | U | V | P | XP | I | RI | KC |`

`TR]`

Or

`ddlgen -v`

defncopy

Also use for `defncopy_r`, `defncopy_dce`.

`defncopy`  
`[-X]`  
`[-a display_charset]`  
`[-I interfaces_file]`  
`[-J [client_charset]]`  
`[-K keytab_file]`  
`[-P password]`  
`[-R remote_server_principal]`  
`[-S [server_name]]`  
`[-U username]`  
`[-V security_options]`  
`[-Z security_mechanism]`  
`[-z language]`  
`{ in file_name database_name |`  
`out file_name database_name [owner.]object_name`  
`[[owner.]object_name...] }`

Or

`defncopy -v`

dscp

Also use for `dscp_r`, `dscp_dce`.

`dscp [-p]`

or

dscp -v

To exit from dscp:

quit

or

exit

dsedit

Also use for dsedit\_r, dsedit\_dce.

dsedit

or

dsedit -v

extractjava

extrjava.exe in Windows NT.

extractjava (extrjava in Windows NT)

```
-j jar_name
-f file_name
[-S server_name]
[-U user_name]
[-P password]
[-D database_name]
[-l interfaces_file]
[-a display_charset]
[-J client_charset]
[-z language]
[-t timeout]
[-v]
```

Or

extractjava -v

installjava

instjava.exe in Windows NT.

```
installjava
-f file_name
[ -new | -update ]
[-j jar_name ]
[-S server_name ]
[-U user_name ]
[-P password ]
[-D database_name ]
[-l interfaces_file ]
[-a display_charset ]
[-J client_charset ]
[-z language ]
[-t timeout ]
[-v]
```



---

Or

installjava -v

isql

Also use for isql\_r, isql\_dce.

```
isql [-b] [-e] [-F] [-p] [-n] [-v] [-X] [-Y] [-Q]
      [-a display_charset]
      [-A packet_size]
      [-c cmdend]
      [-D database]
      [-E editor]
      [-h headers]
      [-H hostname]
      [-i inputfile]
      [-I interfaces_file]
      [-J client_charset]
      [-K keytab_file]
      [-l login_timeout]
      [-m errorlevel]
      [-o outputfile]
      [-P password]
      [-R remote_server_principal]
      [-s colseparator]
      [-S server_name]
      [-t timeout]
      -U username
      [-V [security_options]]
      [-w columnwidth]
      [-z locale_name]
      [-Z security_mechanism]
```

- To terminate a command:  
go
- To clear the query buffer:  
reset
- To call the default editor:  
vi
- To execute an operating system command:  
!! *command*
- To exit from isql:  
quit  
or

exit

---

**Note** [-K *keytab\_file*], [-M *labelname labelvalue*], and [-V *security\_options*] are UNIX only.

---

langinstall

langinst.exe in Windows NT.

```
langinstall
[-S server]
[-U user]
[-I interfaces_file]
[-P password]
[-R release_number]
[-I path_to_interfaces_file]
language
character_set
```

Or

```
langinstall -v
```

optdiag

```
optdiag [binary] [simulate] statistics
{ -i input_file |
  database[.owner].[table[.column] ] ] } [-o output_file] }
[-U user_name]
[-P password]
[-T trace_value]
[-I interfaces_file]
[-S server]
[-v]
[-h]
[-s]
[-z language]
[-J client_character_set]
[-a display_charset]
```

pwdcrypt

pwdcrypt

showserver

showserver

sqldbgr

```
sqldbgr
-U username
-P password
-S host:port
```

sqlloc

```
sqlloc
[-S Server]
[-U User]
[-P Password]
[-s Sybase Dir]
[-I Interfaces file]
[-r Resource file]
```

---

Or

sqlloc -v

sqllocres            sqllocres  
                       [-S *Server*]  
                       [-U *User*]  
                       [-P *Password*]  
                       [-s *Sybase Dir*]  
                       [-l *Interfaces file*]  
                       [-r *Resource file*]

Or

sqllocres -v

sqlsrvr            Windows NT only.

sqlserver [-f] [-g] [-G] [-h] [-H] [-m] [-P] [-q] [-v] [-X]  
           [-a *path\_to\_CAPs\_directive\_file*]  
           [-b *master\_device\_size*]  
           [-c *config\_file\_for\_server*]  
           [-d *device\_name*]  
           [-e *path\_to\_error\_log*]  
           [-i *interfaces\_file\_directory*]  
           [-K *keytab\_file*]  
           [-L *config\_file\_name\_for\_connectivity*]  
           [-M *shared\_memory\_repository\_directory*]  
           [-p *sa\_login\_name*]  
           [-r *mirror\_disk\_name*]  
           [-s *server\_name*]  
           [-T *trace\_flag*]  
           [-u *sa/sso\_name*]  
           [-w *master | model database*]  
           [-y [*password*] ]  
           [-z *page\_size* [ *k* | *K* ] ]

sqlupgrade            sqlupgrade  
                       [-s *Sybase Dir*]  
                       [-r *Resource File*]

Or

sqlupgrade -v

sqlupgraderes        sqlupgraderes  
                       [-s *Sybase Dir*]  
                       [-r *Resource File*]

Or

sqlupgraderes -v

srvbuild            srvbuild  
                       [-s *sybase\_dir*]

```

        [-l interfaces_file]
        [-r resource_file]

Or

    srvbuild -v
srvbuildres    srvbuildres
                [-ssybase_dir]
                [-linterfaces_file]
                [-rresource_file]

Or

    srvbuildres -v
startserver    startserver [[-f runserverfile] [-m]] ...
sybmigrate     sybmigrate [-v ] [-h ] [-f ]
                [-D 1 | 2 | 3 | 4 ]
                [-l interfaces file ]
                [-r input resource file ]
                [-m setup | migrate | validate | report ]
                [-rn status | space_est | repl | diff | password ]
                [-l log file ]
                [-t output template resource file ]
                [-J client_charset ]
                [-z language ]
                [-T trace_flags ]
                [-Tase trace flags ]
                [-f ]

xpserver       xpserver -S XP_Server
                xpserver
                -SXP_Server
                [-linterfaces_file]
                [-ppriority]
                [-sstack_size]
                [-u]
                [-v]
                [-x]

```