

SYBASE®

リファレンス・マニュアル：テーブル

Adaptive Server® Enterprise

15.5

ドキュメント ID : DC37455-01-1550-01

改訂 : 2009 年 11 月

Copyright © 2010 by Sybase, Inc. All rights reserved.

このマニュアルは Sybase ソフトウェアの付属マニュアルであり、新しいマニュアルまたはテクニカル・ノートで特に示されないかぎり、後続のリリースにも付属します。このマニュアルの内容は予告なしに変更されることがあります。このマニュアルに記載されているソフトウェアはライセンス契約に基づいて提供されるものであり、無断で使用することはできません。

このマニュアルの内容を弊社の書面による事前許可を得ずに、電子的、機械的、手作業、光学的、またはその他のいかなる手段によっても、複製、転載、翻訳することを禁じます。

マニュアルの注文

マニュアルの注文を承ります。ご希望の方は、サイベース株式会社営業部または代理店までご連絡ください。マニュアルの変更は、弊社の定期的なソフトウェア・リリース時のみ提供されます。

Sybase の商標は、**Sybase trademarks ページ** (<http://www.sybase.com/detail?id=1011207>) で確認できます。Sybase およびこのリストに掲載されている商標は、米国法人 Sybase, Inc. の商標です。® は、米国における登録商標であることを示します。

Java および Java 関連の商標は、米国およびその他の国における Sun Microsystems, Inc. の商標または登録商標です。

Unicode と Unicode のロゴは、Unicode, Inc. の登録商標です。

IBM および Tivoli は、International Business Machines Corporation の米国およびその他の国における登録商標です。

このマニュアルに記載されている上記以外の社名および製品名は、当該各社の商標または登録商標の場合があります。

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.

目次

はじめに	vii
第 1 章	システム・テーブル 1
システム・テーブルのロケーション	1
master 内のシステム・テーブル	1
sybsecurity 内のシステム・テーブル	2
sybssystemdb 内のシステム・テーブル	2
すべてのデータベース内のシステム・テーブル	3
sybdiagdb データベースの概要	4
syblicenseslog テーブルの概要	4
Cluster Edition でのシステム・テーブルの使用	4
timestamp カラム	4
変更された identity 値	5
偽のテーブルのマテリアライズの制御	5
システム・テーブル使用上の規則	6
システム・テーブルのパーミッション	7
システム・テーブルに使用されるロック・スキーム	7
予約済みカラム	8
システム・テーブルの更新	8
システム・テーブル上のトリガ	8
syblicenseslog	9
sysalternates	10
sysaltusages	11
sysattributes	13
sysauditoptions	15
sysaudits_01 ~ sysaudits_08	16
syscharsets	18
syscolumns	19
syscomments	21
sysconfigures	23
sysconstraints	25
syscoordinations	26
syscurconfigs	27
sysdatabases	29
sysdepends	32
sysdevices	33

sysencryptkeys	35
sysengines	37
sysgams	38
sysindexes.....	39
sysinstances.....	42
sysjars	43
syskeys.....	44
syslanguages	45
syslisteners.....	46
syslocks.....	47
sysloginroles.....	49
syslogins.....	50
syslogs	52
syslogshold.....	53
sysmessages.....	54
sysmonitors	55
sysobjects.....	56
sysoptions	61
syspartitionkeys.....	62
syspartitions	63
sysprocedures	65
sysprocesses.....	66
sysprotects	69
sysquerymetrics	70
sysqueryplans	72
sysreferences	73
sysremotelogins	74
sysresourcelimits.....	75
sysroles	76
syssecmechs.....	77
syssegments	78
sysservers	79
syssessions	81
syslices	82
sysrvroles	83
sysstatistics	84
systabstats	85
systhresholds	87
systemranges.....	89
systransactions.....	90
systypes	93
sysusages	96
sysusermessages.....	97
sysusers	98
sysxtypes.....	99

第 2 章	dbccdb のテーブル	101
	dbccdb 作業領域	101
	dbccdb ログ	103
	dbcc_config	104
	dbcc_counters	105
	dbcc_exclusions	106
	dbcc_fault_params	107
	dbcc_faults	108
	dbcc_operation_log	109
	dbcc_operation_results	110
	dbcc_types	111
第 3 章	モニタリング・テーブル	119
	monCachedObject	120
	monCachePool	121
	monCachedProcedures	122
	monCachedStatement	123
	monCIPC	126
	monCIPCEndpoints	127
	monCIPCLinks	128
	monCIPCMesh	129
	monCLMObjectActivity	131
	monClusterCacheManager	133
	monCMSFailover	134
	monDataCache	135
	monDBRecovery	136
	monDBRecoveryLRTypes	138
	monDeadLock	139
	monDeviceIO	141
	monEngine	142
	monErrorLog	144
	monFailoverRecovery	145
	monIOQueue	146
	monLicense	147
	monLocks	148
	monLogicalCluster	150
	monLogicalClusterAction	152
	monLogicalClusterInstance	153
	monLogicalClusterRoute	154
	monNetworkIO	155
	monOpenDatabases	156
	monOpenObjectActivity	157
	monOpenPartitionActivity	160
	monPCIBridge	163
	monPCIEngine	163
	monPCISlots	164

monPCM	165
monProcedureCache	167
monProcedureCacheMemoryUsage	168
monProcedureCacheModuleUsage	169
monProcess	170
monProcessActivity	171
monProcessLookup	172
monProcessMigration	172
monProcessNetIO	173
monProcessObject	174
monProcessProcedures	175
monProcessSQLText	176
monProcessStatement	177
monProcessWaits	178
monProcessWorkerThread	179
monState	180
monStatementCache	181
monSysLoad	182
monSysPlanText	183
monSysSQLText	184
monSysStatement	185
monSysWaits	187
monSysWorkerThread	188
monTableColumns	189
monTableParameters	190
monTables	191
monTableTransfer	192
monTempdbActivity	193
monWaitClassInfo	194
monWaitEventInfo	195
monWorkload	196
monWorkloadPreview	197
monWorkloadProfile	198
monWorkloadRaw	199
第 4 章 sybpcidb のテーブル	201
pca_jre_arguments	202
pca_jre_directives	203
pci_arguments	204
pci_directives	205
pci_slotinfo	206
pci_slot_syscalls	207
索引	209

はじめに

『ASE リファレンス・マニュアル』は、Sybase® Adaptive Server® Enterprise と Transact-SQL® 言語について説明する 4 巻構成のマニュアルです。

- 『ビルディング・ブロック』では、データ型、組み込み関数、グローバル変数、式と識別子、予約語、SQLSTATE エラーなど、Transact-SQL の各要素について説明します。Transact-SQL を使用する前に、ビルディング・ブロックの動作とビルディング・ブロックが Transact-SQL 文の結果にどのように影響するかを理解する必要があります。
- 『コマンド』では、文の作成に使用する Transact-SQL コマンドについてのリファレンス情報が記載されています。
- 『プロシージャ』では、システム・プロシージャ、カタログ・ストアド・プロシージャ、拡張ストアド・プロシージャ、dbcc ストアド・プロシージャについてのリファレンス情報が記載されています。すべてのプロシージャは、Transact-SQL 文を使用して作成されます。
- 『テーブル』では、使用しているサーバ、データベース、ユーザ、サーバの他の詳細についての情報を格納するシステム・テーブルに関するリファレンス情報が記載されています。dbccdb データベースと dbccalt データベースのテーブルについても説明します。

対象読者

『ASE リファレンス・マニュアル』は、リファレンスのツールとして Transact-SQL のすべてのレベルのユーザを対象としています。

このマニュアルの内容

- 「**第 1 章 システム・テーブル**」では、master データベース、auditing データベース、およびその他すべてのユーザ・データベース (pubs2 など) のシステム・テーブルに関する情報を記載します。またこの章では、syblicenseslog についても説明します。これはシステム・データベースではありませんが、ライセンスに関する情報を提供します。
- 「**第 2 章 dbccdb のテーブル**」では、dbccdb および dbccalt データベースのテーブルについて説明します。
- 「**第 3 章 モニタリング・テーブル**」では、モニタリング・テーブルについて説明します。

関連マニュアル

Adaptive Server® Enterprise には次のマニュアルが用意されています。必要に応じて参照してください。

- 使用しているプラットフォームの『リリース・ノート』 - マニュアルには記載できなかった最新の情報が記載されています。
このリリース・ノートの最新バージョン (英語版) を入手できます。製品の CD がリリースされた後、製品またはマニュアルに関する重要な情報が追加されているかを確認するには、Sybase® Product Manuals Web サイトを使用してください。
- 使用しているプラットフォームの『インストール・ガイド』 - すべての Adaptive Server および関連する Sybase 製品のインストール、アップグレード、設定の手順について説明しています。
- 『新機能ガイド』 - Adaptive Server の新しい機能について説明しています。また、新しい機能をサポートするためのシステム変更や、既存のアプリケーションに影響を与える可能性がある変更についても説明しています。
- 『Active Messaging ユーザーズ・ガイド』 - Active Messaging を使用して、Adaptive Server Enterprise データベースでトランザクション (データ変更) を取得し、外部アプリケーションにイベントとしてリアルタイムで渡す方法について説明しています。
- 『コンポーネント統合サービス・ユーザーズ・ガイド』 - コンポーネント統合サービスを使用して、リモートの Sybase データベースおよび Sybase 以外のデータベースに接続する方法について説明しています。
- 使用しているプラットフォームの『設定ガイド』 - 特定の設定作業の手順について説明しています。
- 『用語解説』 - Adaptive Server マニュアルで使用されている技術用語について説明しています。
- 『Historical Server ユーザーズ・ガイド』 - Historical Server を使用して、Adaptive Server のパフォーマンス情報を入手する方法について説明しています。
- 『Adaptive Server Enterprise における Java』 - Adaptive Server データベースで Java クラスをデータ型、関数、ストアド・プロシージャとしてインストールして使用する方法について説明しています。
- 『Job Scheduler ユーザーズ・ガイド』 - コマンド・ラインまたはグラフィカル・ユーザ・インタフェース (GUI) を使用して、ローカルまたはリモートの Adaptive Server でジョブのインストール、設定、作成、スケジュールを行う方法について説明しています。
- 『マイグレーション技術ガイド』 - 別のバージョンの Adaptive Server にマイグレートするための方法とツールについて説明しています。
- 『Monitor Client Library プログラマーズ・ガイド』 - Adaptive Server のパフォーマンス・データにアクセスする Monitor Client Library アプリケーションの記述方法について説明しています。

- 『Monitor Server ユーザーズ・ガイド』 – Monitor Server を使用して、Adaptive Server のパフォーマンス統計を取得する方法について説明しています。
- 『モニタリング・テーブル・ダイヤグラム』 – モニタリング・テーブルと、そのエンティティの関係をポスター形式で図解しています。フル・サイズのダイヤグラムは印刷版だけで参照できます。コンパクト版は PDF 形式で参照できます。
- 『パフォーマンス&チューニング・シリーズ』 – Adaptive Server で最高のパフォーマンスを実現するためのチューニング方法について説明しています。このマニュアルは以下の 7 冊に分かれています。
 - 『基本』 – Adaptive Server のパフォーマンスに関する問題の理解と調査の基本について説明しています。
 - 『統計的分析によるパフォーマンスの向上』 – Adaptive Server で統計情報がどのように保存され、表示されるかについて説明しています。また、`set statistics` コマンドを使用して、サーバの統計情報を分析する方法について説明しています。
 - 『ロックと同時実行制御』 – ロック・スキームを使用してパフォーマンスを向上させる方法と、同時実行性を最小限に抑えるようにインデックスを選択する方法について説明しています。
 - 『sp_sysmon による Adaptive Server の監視』 – `sp_sysmon` を使用してパフォーマンスをモニタリングする方法について説明しています。
 - 『モニタリング・テーブル』 – Adaptive Server のモニタリング・テーブルに統計情報や診断情報を問い合わせる方法について説明しています。
 - 『物理データベースのチューニング』 – データの物理的配置、データに割り付けられた領域、テンポラリー・データベースの管理方法について説明しています。
 - 『クエリ処理と抽象プラン』 – オプティマイザがクエリを処理する方法と、抽象プランを使用してオプティマイザのプランの一部を変更する方法について説明しています。
- 『クイック・リファレンス・ガイド』 – コマンド、関数、システム・プロシージャ、拡張システム・プロシージャ、データ型、ユーティリティの名前と構文の包括的な一覧表を記載したポケット版 (PDF 版は通常サイズ) のマニュアルです。

-
- 『ASE リファレンス・マニュアル』－ 詳細な Transact-SQL® 情報を記載しています。このマニュアルは以下の 4 冊に分かれています。
 - 『ビルディング・ブロック』－ データ型、関数、グローバル変数、式、識別子とワイルドカード、予約語について説明しています。
 - 『コマンド』－ コマンドについて説明しています。
 - 『プロシージャ』－ システム・プロシージャ、カタログ・ストアド・プロシージャ、システム拡張ストアド・プロシージャ、dbcc ストアド・プロシージャについて説明しています。
 - 『テーブル』－ システム・テーブル、モニタリング・テーブル、dbcc テーブルについて説明しています。
 - 『システム管理ガイド』－
 - 『第 1 巻』－ 設定パラメータ、リソースの問題、文字セット、ソート順、システムの問題の診断方法に関する説明を含め、システム管理の基本の概要について説明しています。『第 1 巻』の後半は、セキュリティ管理に関する詳細な説明です。
 - 『第 2 巻』－ 物理的なリソースの管理、デバイスのミラーリング、メモリとデータ・キャッシュの設定、マルチプロセッサ・サーバとユーザ・データベースの管理、データベースのマウントとマウント解除、セグメントの作成と使用、reorg コマンドの使用、データベース一貫性の検査方法についての手順とガイドラインを説明しています。『第 2 巻』の後半では、システムとユーザ・データベースをバックアップおよびリストアする方法について説明しています。
 - 『システム・テーブル・ダイアグラム』－ システム・テーブルと、そのエンティティとの関係をポスター形式で図解しています。フル・サイズのダイアグラムは印刷版だけで参照できます。コンパクト版は PDF 形式で参照できます。
 - 『Transact-SQL ユーザーズ・ガイド』－ リレーショナル・データベース言語の拡張版である Sybase の Transact-SQL について説明しています。まだ経験の浅いデータベース管理システムのユーザは、このマニュアルをガイドブックとして使用してください。pubs2 および pubs3 サンプル・データベースの詳細も説明しています。
 - 『トラブルシューティング：エラー・メッセージと詳細な解決方法』－ 発生する可能性のある問題について、トラブルシューティング手順を説明しています。このマニュアルで取り上げられている問題は、Sybase 製品の保守契約を結んでいるサポート・センタに最も頻繁に寄せられるものです。
 - 『暗号化カラム・ユーザーズ・ガイド』－ Adaptive Server を使用して暗号化カラムを設定し、使用方法について説明しています。
 - 『インメモリ・データベース・ユーザーズ・ガイド』－ インメモリ・データベースの設定および使用方法について説明しています。

- 『Adaptive Server 分散トランザクション管理機能の使用』－分散トランザクション処理環境での Adaptive Server DTM 機能の設定、使用、トラブルシューティングについて説明しています。
- 『IBM® Tivoli® Storage Manager と Backup Server の使用』－IBM Tivoli Storage Manager を設定および使用して Adaptive Server のバックアップを作成する方法について説明しています。
- 『高可用性システムにおける Sybase フェールオーバの使用』－Sybase のフェールオーバ機能を使用して、Adaptive Server を高可用性システムのコンパニオン・サーバとして設定する方法について説明しています。
- 『Unified Agent および Agent Management Console』－Unified Agent について説明しています。Unified Agent は、分散 Sybase リソースを管理、モニタ、制御するためのランタイム・サービスを提供します。
- 『ASE ユーティリティ・ガイド』－オペレーティング・システム・レベルで実行される `isql` および `bcp` などの、Adaptive Server のユーティリティ・プログラムについて説明しています。
- 『Web Services ユーザーズ・ガイド』－Adaptive Server 用の Web サービスの設定、使用、トラブルシューティング方法について説明しています。
- 『XA インタフェース統合ガイド for CICS, Encina, TUXEDO』－X/Open XA トランザクション・マネージャを備えた Sybase DTM XA インタフェースを使用する方法について説明しています。
- 『Adaptive Server Enterprise における XML サービス』－データベースに XML 機能を導入する、Sybase ネイティブの XML プロセッサと Sybase Java ベースの XML のサポートについて、また XML サービスで使用できるクエリとマッピング用の関数について説明しています。

その他の情報

Sybase Getting Started CD、SyBooks™ CD、Sybase Product Manuals Web サイトを利用すると、製品について詳しく知ることができます。

- Getting Started CD には、PDF 形式のリリース・ノートとインストール・ガイド、SyBooks CD に含まれていない他のマニュアルや更新情報が収録されています。この CD は製品のソフトウェアに同梱されています。Getting Started CD に収録されているマニュアルを参照または印刷するには、Adobe Acrobat Reader が必要です (CD 内のリンクを使用して Adobe の Web サイトから無料でダウンロードできます)。
- SyBooks CD には製品マニュアルが収録されています。この CD は製品のソフトウェアに同梱されています。Eclipse ベースの SyBooks ブラウザを使用すれば、使いやすい HTML 形式のマニュアルにアクセスできます。
一部のマニュアルは PDF 形式で提供されています。これらのマニュアルは SyBooks CD の PDF ディレクトリに収録されています。PDF ファイルを開いたり印刷したりするには、Adobe Acrobat Reader が必要です。

SyBooks をインストールして起動するまでの手順については、Getting Started CD の『SyBooks インストール・ガイド』、または SyBooks CD の *README.txt* ファイルを参照してください。

- Sybase Product Manuals Web サイトは、SyBooks CD のオンライン版であり、標準の Web ブラウザを使用してアクセスできます。また、製品マニュアルのほか、EBFs/Updates、Technical Documents、Case Management、Solved Cases、ニュース・グループ、Sybase Developer Network へのリンクもあります。

Technical Library Product Manuals Web サイトにアクセスするには、Product Manuals (<http://www.sybase.com/support/manuals/>) にアクセスしてください。

Web 上の Sybase 製品の動作確認情報

Sybase Web サイトの技術的な資料は頻繁に更新されます。

❖ 製品認定の最新情報にアクセスする

- 1 Web ブラウザで Technical Documents を指定します。
(<http://www.sybase.com/support/techdocs/>)
- 2 [Certification Report] をクリックします。
- 3 [Certification Report] フィルタで製品、プラットフォーム、時間枠を指定して [Go] をクリックします。
- 4 [Certification Report] のタイトルをクリックして、レポートを表示します。

❖ コンポーネント認定の最新情報にアクセスする

- 1 Web ブラウザで Availability and Certification Reports を指定します。
(<http://certification.sybase.com/>)
- 2 [Search By Base Product] で製品ファミリーとベース製品を選択するか、[Search by Platform] でプラットフォームとベース製品を選択します。
- 3 [Search] をクリックして、入手状況と認定レポートを表示します。

❖ Sybase Web サイト (サポート・ページを含む) の自分専用のビューを作成する

MySybase プロファイルを設定します。MySybase は無料サービスです。このサービスを使用すると、Sybase Web ページの表示方法を自分専用カスタマイズできます。

- 1 Web ブラウザで Technical Documents を指定します。
(<http://www.sybase.com/support/techdocs/>)
- 2 [MySybase] をクリックし、MySybase プロファイルを作成します。

Sybase EBF とソフトウェア・メンテナンス

❖ EBF とソフトウェア・メンテナンスの最新情報にアクセスする

- 1 Web ブラウザで Sybase Support Page を指定します。
(<http://www.sybase.com/support>)
- 2 [EBFs/Maintenance] を選択します。MySybase のユーザ名とパスワードを入力します。
- 3 製品を選択します。
- 4 時間枠を指定して [Go] をクリックします。EBF/Maintenance リリースの一覧が表示されます。

鍵のアイコンは、「Technical Support Contact」として登録されていないため、一部の EBF/Maintenance リリースをダウンロードする権限がないことを示しています。未登録でも、Sybase 担当者またはサポート・コンタクトから有効な情報を得ている場合は、[Edit Roles] をクリックして、「Technical Support Contact」の役割を MySybase プロファイルに追加します。
- 5 EBF/Maintenance レポートを表示するには [Info] アイコンをクリックします。ソフトウェアをダウンロードするには製品の説明をクリックします。

表記規則

次の項では、このマニュアルで使用されている表記について説明します。

SQL は自由な形式の言語で、1 行内のワード数や、改行の仕方に規則はありません。このマニュアルでは、読みやすくするため、例や構文を文の句ごとに改行しています。複数の部分からなり、2 行以上にわたる場合は、字下げしています。複雑なコマンドの書式には、修正された BNF (Backus Naur Form) 記法が使用されています。

表 1 に構文の規則を示します。

表 1: このマニュアルでのフォントと構文規則

要素	例
コマンド名、プロシージャ名、ユーティリティ名、その他のキーワードは sans serif フォントで表記する。	<code>select</code> <code>sp_configure</code>
データベース名とデータ型は sans serif フォントで表記する。	<code>master</code> データベース
ファイル名、変数、パス名は斜体で表記する。	システム管理ガイド <i>sql.ini</i> ファイル <i>column_name</i> <i>SSYBASE/ASE</i> ディレクトリ
変数 (ユーザが入力する値を表す語) がクエリまたは文の一部である場合は Courier フォントの斜体で表記する。	<code>select column_name</code> <code>from table_name</code> <code>where search_conditions</code>
カッコはコマンドの一部として入力する。	<code>compute row_aggregate (column_name)</code>

要素	例
2つのコロンと等号は、構文がBNF表記で記述されていることを示す。この記号は入力しない。「~と定義されている」ことを意味する。	::=
中カッコは、その中のオプションを1つ以上選択しなければならないことを意味する。コマンドには中カッコは入力しない。	{cash, check, credit}
角カッコは、オプションを選択しても省略してもよいことを意味する。コマンドには角カッコは入力しない。	[cash check credit]
中カッコまたは角カッコの中のカンマで区切られたオプションをいくつでも選択できることを意味する。複数のオプションを選択する場合には、オプションをカンマで区切る。	cash, check, credit
パイプまたは縦線は複数のオプションのうち1つだけを選択できることを意味する。	cash check credit
省略記号 (...) は、直前の要素を必要な回数だけ繰り返し指定できることを意味する。	buy thing = price [cash check credit] [, thing = price [cash check credit]]... この例では、製品 (thing) を少なくとも1つ購入 (buy) し、価格 (price) を指定する必要があります。支払方法を選択できる。角カッコで囲まれた項目の1つを選択する。追加品目を、必要な数だけ購入することもできる。各 buy に対して、購入した製品 (thing)、価格 (price)、オプションで支払方法 (cash, check, credit のいずれか) を指定します。

- 次は、オプション句のあるコマンドの構文の例です。

```
sp_dropdevice [device_name]
```

複数のオプションを持つコマンドの例を示します。

```
select column_name
from table_name
where search_conditions
```

構文では、キーワード (コマンド) は通常のフォントで表記し、識別子は小文字で表記します。ユーザが提供するワードは斜体で表記します。

- Transact-SQL コマンドの使用例は次のように表記します。

```
select * from publishers
```

- 次は、コンピュータからの出力例です。

```
pub_id      pub_name                city                state
-----
0736       New Age Books           Boston             MA
0877       Binnet & Hardley        Washington         DC
1389       Algodata Infosystems   Berkeley          CA
```

(3 rows affected)

このマニュアルでは、例に使用する文字はほとんどが小文字ですが、Transact-SQL のキーワードを入力するときは、大文字と小文字は区別されません。たとえば、SELECT、Select、select はすべて同じです。

テーブル名などのデータベース・オブジェクトの大文字と小文字を Adaptive Server が区別するかどうかは、Adaptive Server にインストールされたソート順によって決まります。シングルバイト文字セットを使用している場合は、Adaptive Server のソート順を再設定することによって、大文字と小文字の区別の取り扱い方を変更できます。詳細については、『システム管理ガイド』を参照してください。

アクセシビリティ機能

このマニュアルには、アクセシビリティを重視した HTML 版もあります。この HTML 版マニュアルは、スクリーン・リーダーで読み上げる、または画面を拡大表示するなどの方法により、その内容を理解できるよう配慮されています。

Adaptive Server HTML マニュアルは、連邦リハビリテーション法第 508 条のアクセシビリティ規定に準拠していることがテストにより確認されています。第 508 条に準拠しているマニュアルは通常、World Wide Web Consortium (W3C) の Web サイト用ガイドラインなど、米国以外のアクセシビリティ・ガイドラインにも準拠しています。

注意 アクセシビリティ・ツールを効率的に使用するには、設定が必要な場合もあります。一部のスクリーン・リーダーは、テキストの大文字と小文字を区別して発音します。たとえば、すべて大文字のテキスト (ALL UPPERCASE TEXT など) はイニシャルで発音し、大文字と小文字の混在したテキスト (Mixed Case Text など) は単語として発音します。構文規則を発音するようにツールを設定すると便利かもしれません。詳細については、ツールのマニュアルを参照してください。

Sybase のアクセシビリティに対する取り組みについては、Sybase Accessibility (<http://www.sybase.com/accessibility>) を参照してください。Sybase Accessibility サイトには、第 508 条と W3C 標準に関する情報へのリンクもあります。

不明な点があるときは

Sybase ソフトウェアがインストールされているサイトには、Sybase 製品の保守契約を結んでいるサポート・センタとの連絡担当の方 (コンタクト・パーソン) を決めてあります。マニュアルだけでは解決できない問題があった場合には、担当の方を通して Sybase のサポート・センタまでご連絡ください。



システム・テーブルは、Sybase から提供されます。Adaptive Server バージョン 15.0 のシステム・テーブルは、ほとんどがロー・ロック・テーブルです。ロー・ロック・テーブルでないテーブルについては、各システム・テーブルの説明の中で言及しています。

トピック名	Page
システム・テーブルのロケーション	1
システム・テーブル使用上の規則	6

システム・テーブルのロケーション

システム・テーブルのロケーション

- master データベース
- sybsecurity データベース
- sybssystemdb データベース
- すべてのデータベース

master データベース内のテーブルは、ほとんどがシステム・テーブルです。システム・テーブルの一部は、ユーザ・データベースにも同様に存在します。これらのシステム・テーブルは、`create database` コマンドが実行されると、自動的に作成されます。

master 内のシステム・テーブル

次に示すシステム・テーブルは、master データベースだけにあります。

システム・テーブル	内容
syscharsets	文字セットまたはソート順それぞれにつき 1 つのロー。
sysconfigures	ユーザが設定できる設定パラメータにつき 1 つのロー。
syscurconfigs	Adaptive Server が現在使用している設定パラメータについての情報。
sysdatabases	Adaptive Server 上の各データベースにつき 1 つのロー。
sysdevices	テープ・ダンプ・デバイス、ディスク・ダンプ・デバイス、データベース用のディスク、およびデータベース用のディスク・パーティションそれぞれにつき 1 つのロー。
sysengines	現在オンラインになっている Adaptive Server エンジンごとに 1 つのロー。
syslanguages	サーバが認識している各言語 (us_english を除く) につき 1 つのロー。

システム・テーブル	内容
syslisteners	現在の Adaptive Server が使用しているネットワーク接続のタイプごとに1つのロー。
syslocks	アクティブなロックについての情報。
sysloginroles	システム定義役割を持つサーバ・ログインごとに1つのロー。
syslogins	有効な Adaptive Server ユーザ・アカウントごとに1つのロー。
syslogshold	各データベースの最も古いアクティブなトランザクションと Replication Server® トランザクション・ポイントについての情報。
sysmessages	システム・エラーまたは警告ごとに1つのロー。
sysmonitors	モニタ・カウンタごとに1つのロー。
sysprocesses	サーバの処理についての情報。
sysremotelogins	リモート・ユーザごとに1つのロー。
sysresourceimits	リソース制限ごとに1つのロー。
syssecmechs	Adaptive Server で利用できる各セキュリティ・メカニズムに提供されているセキュリティ・サービスに関する情報。
syssservers	リモート Adaptive Server ごとに1つのロー。
sysssessions	高可用性システム内で Adaptive Server が Sybase フェールオーバー用に設定されている場合にのみ使用される。 sysssessions には、フェールオーバー・プロパティを使用して Adaptive Server に接続するクライアントごとに1つのローが存在する。
sysrvroles	サーバワイドな役割ごとに1つのロー。
systimeranges	名前付き時間範囲ごとに1つのロー。
systransactions	トランザクションごとに1つのロー。
sysusages	データベースに割り当てられたディスクの一部分それぞれにつき1つのロー。

sybsecurity 内のシステム・テーブル

次に示すシステム・テーブルは、sybsecurity データベースだけあります。

システム・テーブル	内容
sysauditoptions	グローバル監査オプションごとに1つのロー。
sysaudits_01 ~ sysaudits_08	監査証跡。監査テーブルには、監査レコードごとに1つのローが含まれている。

監査に関連するすべてのシステム・テーブルは全ページ・ロックです。

sybssystemdb 内のシステム・テーブル

次に示すシステム・テーブルは、sybssystemdb データベースだけにあります。

システム・テーブル	内容
syscoordinations	分散トランザクションのリモート・パティシバントごとに1つのロー。

すべてのデータベース内のシステム・テーブル

次のシステム・テーブルはすべてのデータベースにあります。

システム・テーブル	内容
sysalternates	データベースのユーザにマップされた Adaptive Server ユーザごとに 1 つのロー。
sysattributes	オブジェクトの属性定義ごとに 1 つのロー。
syscolumns	テーブルまたはビューのカラムごとに 1 つのロー、およびプロシージャのパラメータごとに 1 つのロー。
syscomments	ビュー、ルール、デフォルト、トリガ、プロシージャそれぞれにつき 1 つまたは複数のロー。SQL 定義文を示す。
sysconstraints	テーブルまたはカラムに関連する参照制約および検査制約それぞれにつき 1 つのロー。
sysdepends	プロシージャ、ビュー、トリガによって参照されるプロシージャ、ビュー、テーブルそれぞれにつき 1 つのロー。
sysgams	データベース全体のアロケーション・ビットマップ。
sysindexes	クラスタード・インデックスまたはノンクラスタード・インデックスごとに 1 つのロー。インデックスのないテーブルごとに 1 つのロー。text データや image データを持つテーブルごとに追加ロー。
sysjars	データベースに必要な Java アーカイブ (JAR) ファイルごとに 1 つのローを持つ。
syskeys	ユーザが設定するプライマリ・キー、外部キー、共通キーそれぞれにつき 1 つのロー (Adaptive Server では管理されない)。
syslogs	トランザクション・ログ。
sysobjects	テーブル、ビュー、プロシージャ、ルール、トリガ・デフォルト、ログ、およびテンポラリ・オブジェクト (tempdb 内のみ) それぞれにつき 1 つのロー。
syspartitionkeys	パーティション・キーごとに 1 つのロー。
syspartitions	分割されたテーブルまたはインデックスの分割ごとに 1 つのロー。
sysprocedures	ビュー、ルール、デフォルト、トリガ、プロシージャそれぞれにつき 1 つのロー。内部定義を示す。
sysprotects	ユーザのパーミッション情報。
sysquerymetrics	永続的なカタログで集計履歴クエリ情報を収集する。sysquerymetrics は、ビューであり、テーブルではない。
sysqueryplans	抽象クエリ・プランおよび SQL テキスト。
sysreferences	テーブルまたはカラムに宣言された参照整合性制約につき 1 つのロー。
sysroles	ローカルのデータベース・グループにマップされるサーバワイドな役割。
syssegments	各セグメント (ディスクの一部の集まりに名前を付けたもの) につき 1 つのロー。
syslices	旧式のテーブルであり、アップグレード時にのみ使用する。Adaptive Server バージョン 15.0 より前の名称は syspartitions。
sysstatistics	ユーザ・テーブル上の各インデックス・カラムに対して 1 つ以上のロー。インデックスされていないカラムに対してローを含むこともあります。
systabstats	テーブルごとに 1 つのロー、各ノンクラスタード・インデックスに対して 1 ロー追加する。
systhresholds	データベースに定義されたスレッシュホールドにつき 1 つのロー。
systypes	システム定義データ型およびユーザ定義データ型それぞれにつき 1 つのロー。

システム・テーブル	内容
sysusermessages	各ユーザ定義メッセージにつき1つのロー。
sysusers	データベースに許可されているユーザにつき1つのロー。
sysxtypes	各拡張 Java-SQL データ型につき1つのロー。ロー・レベルのロックを使用する。

sybdiagdb データベースの概要

Sybase 製品の保守契約を結んでいるサポート・センタは、デバッグのために、システム上に **sybdiagdb** データベースを作成することがあります。このデータベースは、テクニカル・サポートが使用する診断用の設定データを保持します。

syblicenseslog テーブルの概要

syblicenseslog テーブルについては、[syblicenseslog \(9 ページ\)](#) を参照してください。技術的にはシステム・テーブルではありませんが、Adaptive Server のシャットダウンに関連するライセンス情報について調べる必要があることが考えられます。

Cluster Edition でのシステム・テーブルの使用

この項では Cluster Edition でシステム・テーブルに加えられた一般的な変更内容について説明します。それぞれのテーブルの変更内容は、表見出しの下にリストされています。

timestamp カラム

Adaptive Server では、テーブルに **timestamp** カラムがある場合、その値はローが変更されるたびに更新されます。クライアント・アプリケーションは、この機能を利用して、「オプティミスティック・ロック」と呼ばれるアクセス・メソッドを使用してローに加えられた変更内容を検出します。**timestamp** カラムの値は、データベース内でユニークな値です。ただし、Cluster Edition では、**timestamp** カラムの値が複数のテーブルにわたってデータベース内で昇順になるとは限りませんが、テーブルごとに昇順になることは保証されています。

変更された identity 値

Cluster Edition の identity カラムの動作は、Adaptive Server の Clustered Edition 以外の Edition の identity カラムの動作とは異なります。Cluster Edition では、identity 値は必ずユニークですが、パフォーマンス上の理由から値が単調に増加しない場合もあります。

Cluster Edition 以外の Adaptive Server では、insert がメモリから次の値にアクセスするときのディスク I/O を削減するために、identity 値のセットがメモリに記録されています。Cluster Edition では、同じサイズの set がメモリに記録されていますが、set は複数のクラスタ・インスタンスに分割されています。identity set のサイズが 250000 である 2 インスタンス・クラスタでは、最初のインスタンスは値 {1, 2, 3 ...} を挿入し、2 番目のインスタンスは値 {125000, 125001, 125002 ...} を挿入します。

next-identity 関数は、next-identity が実行されたインスタンスからテーブルの次の identity 値をレポートします。たとえば、next-identity は、インスタンス 1 に対して 4 を返し、インスタンス 2 に対して 125003 を返します。

Cluster Edition で焼き付けサイズと焼き付け動作に変更がないため、identity-burn-max は、ノンクラスタード Adaptive Server の場合と同様に動作します。

偽のテーブルのマテリアライズの制御

sp_who や sp_lock など、ストアド・プロシージャの中には、sysprocesses や syslocks などの偽のテーブルから読み込むものがあります。偽のテーブルのローはディスクに格納されていないため、偽のテーブルは共有ディスク・クラスタの共有データの動作の例外であり、特別な機能が適用されます。

set system_view コマンドを使用すると、クラスタ内で偽のテーブルのクエリがローカル・インスタンスからローを返すか、すべてのインスタンスからローを返すかを制御できます。set system_view は、セッションレベルのコマンドです。また、set system_view は、モニタリング・テーブルのマテリアライズも制御します。

論理クラスタ・レベルでのデフォルトのシステム・ビューの設定については、『Cluster ユーザーズ・ガイド』を参照してください。

デフォルトでは、Adaptive Server は、ローカル・インスタンスのみからローを取得します。

- 偽のテーブルのクエリがすべてのインスタンスのローをマテリアライズするように指定するには、cluster オプションを使用します。次に例を示します。

```
set system_view cluster
```

- 偽のテーブルのクエリがローカル・インスタンスのローをマテリアライズするように指定するには、instance オプションを使用します。次に例を示します。

```
set system_view instance
```

現在の `system_view` 設定を取得するには、`@@system_view` グローバル変数を選択します。

Adaptive Server は、これらの偽のテーブルのクラスタワイドなマテリアライズをサポートしています。

- `sysprocesses`
- `syslocks`
- `sysengines`
- `syslisteners`
- `sysmonitors`
- `syssechmechs`
- `syscurconfigs`

注意 `sysinstances` は、`system_view` の設定にかかわらず、常にクラスタワイドなマテリアライズに設定されています。

システム・テーブル使用上の規則

この項では、システム・テーブルについての規則、制限、使用方法について説明します。

注意 デフォルトでは、カラムは NOT NULL として定義されています。null 入力可能なカラムについては、“null” キーワードの記述があり、このマニュアルではテーブルのカラムの説明に示されています。

システム・テーブルのパーミッション

システム・テーブルを使用するためのパーミッションは、他のテーブルに対するパーミッションと同様に、データベース所有者が制御できます。デフォルトでは、Adaptive Server をインストールするとき、`installmodel` スクリプトによって、“public” (全ユーザ) にテーブル内のほとんどのフィールドに対する `select` アクセスが付与されます。代わりに、Adaptive Server が新しいデータベースを構築するときに、これらのシステム・テーブルのデフォルト・パーミッションが割り当てられます。ただし、`sysssrvroles` など、システム・テーブルによってはアクセスできないものがあり、他のシステム・テーブルでもアクセスできないフィールドを持つものがあります。たとえば、デフォルトではすべてのユーザが、`audflags` 以外の `sysobjects` のすべてのカラムを選択することができます。詳細については、『システム管理ガイド 第 1 巻』を参照してください。

```
sp_helprotect system_table_name
```

たとえば、`master` の `sysssrvroles` のパーミッションを確認するには、次のコマンドを実行します。

```
use master
go
sp_helprotect sysssrvroles
go
```

システム・テーブルに使用されるロック・スキーム

Adaptive Server の全ページ・ロック・スキームでは、データ・ページ上およびインデックス・ページ上でロックが取得されます。詳細については、『パフォーマンス&チューニング・シリーズ:ロックと同時実行制御』を参照してください。

以下のシステム・テーブルは全ページ・ロックを使用し、これ以外のシステム・テーブルは、データロー・ロックを使用します。

- `sysusermessages`
- `syslices`
- `sysmessages`

また、以下のシステム・テーブルは、外見上は全ページ・ロックを使用した「偽」の (つまりロー指向でない) カタログです。

- `syslogs`
- `sysgams`
- `sysprocesses`
- `syslocks`
- `syscurconfigs`
- `syssecmechs`

- sysmonitors
- sysengines
- systestlog
- syslisteners
- syslogshold

予約済みカラム

カラムの説明の欄に「予約済み」という言葉が入っている場合、Adaptive Server が現在使用していないカラムを意味しています。

システム・テーブルの更新

データベース所有者であっても、システム・テーブルを直接更新することはできません。代わりに、Adaptive Server には、通常必要なシステム・テーブルの更新と追加ができるシステム・プロシージャが用意されており、それらを使用する必要があります。

システム・テーブルに修正が必要な場合に、システム・プロシージャ以外の方法で修正せざるを得ないときには、システム・テーブルを直接更新できます。直接更新を行えるようにするには、システム・セキュリティ担当者がシステム・プロシージャ `sp_configure` を使用して、`allow updates to system tables` という設定パラメータを再設定してください。詳細については、『システム管理ガイド』を参照してください。

システム・テーブル上のトリガ

システム・テーブルにはトリガを作成できません。システム・テーブルにトリガを作成しようとすると、Adaptive Server はエラー・メッセージを返し、トリガの作成を取り消します。

syblicenseslog

master データベースのみ

説明

syblicenseslog には、24 時間を単位として Adaptive Server で使用されたライセンスの最大数の更新ごとに 1 つのローがあります。syblicenseslog は、24 時間ごとに更新されます。Adaptive Server が停止されると、停止する前に License Use Manager が syblicenseslog で現在使用されているライセンス数をログに記録します。Adaptive Server の起動時に、24 時間が開始されます。

注意 syblicenseslog はシステム・テーブルではありません。タイプは“U”、オブジェクト ID は 256 以上です。

カラム

syblicenseslogs のカラムは次のとおりです。

名前	データ型	説明
status	smallint	使用されるライセンスの最大数のステータス。次のいずれか。 <ul style="list-style-type: none"> • 0 = ライセンス数は超過していない。 • 1 = ライセンス数は超過している。 • -1 = ハウスキーピング機能はライセンス数のモニタをできない。
logtime	datetime	ログが書き込まれた日時。
maxlicenses	int	24 時間の間に使用されたライセンスの最大数。

sysalternates

すべてのデータベース

説明

sysalternates には、現在使用しているデータベースのユーザにマップされた (またはエイリアスが設定された) Adaptive Server のユーザごとに、1 つのローが含まれます。ユーザがデータベースにアクセスしようとする時、Adaptive Server は、**sysusers** 内で有効な **uid** エントリを検索します。有効なエントリが見つからなかった場合、**sysalternates.suid** 内を検索します。そこでユーザの **suid** が見つかった場合、ユーザは **sysalternates.altsuid** にリストされている **suid** を所有するデータベース・ユーザとして扱われます。

カラム

sysalternates のカラムは次のとおりです。

名前	データ型	説明
suid	int	マップされているユーザのサーバ・ユーザ ID。
altsuid	int	別のユーザにマップされるユーザのサーバ・ユーザ ID。

インデックス

- **suid** カラムに、ユニーク・クラスタード・インデックスが作成されています。

sysaltusages

スクラッチ・データベース

説明

sysaltusages システム・テーブルは、アーカイブ・データベースのページ番号を、データベース・ダンプとそのストライプ内、または変更済みページ・セクション内の実際のページにマップします。ただし、従来のデータベースの **sysusages** テーブルとは異なり、**sysaltusages** テーブルはデータベースのすべての論理ページをマップするわけではありません。**sysaltusages** は、次のページをマップします。

- データベース・ダンプに格納されたページ
- 変更されたために、変更済みページ・セクションに移動されたページ

『システム管理ガイド 第 2 巻』の「第 14 章 アーカイブ・データベースへのアクセス」を参照してください。

カラム

sysaltusages のカラムは次のとおりです。

名前	データ型	説明
dbid	int	アーカイブ・データベースのデータベース ID。
altsuid	int	ページの物理的に連続したブロックが存在するアーカイブ・データベース・セグメントのロケーション。 location カラムでは、値 5 および 6 はそのロケーションがデータベース・ダンプ、トランザクション・ログ・ダンプ、またはそのストライプにあることを示し、値 7 または 8 はそのロケーションが変更済みページ・セクションにあることを示す。値 4 は、物理的に使用できないページのギャップを埋めるために使用される。
lstart		物理的に連続するページのブロックが開始する論理ページ番号。
start		
size		物理的に連続するページのブロックにある論理ページの数。
vstart		vdevno により指定されたデバイス上のページの連続するブロックが開始するオフセット。
vdevno		ページの連続するブロックが存在するデバイス番号。
segmap		ページのこのブロックが割り付けられたセグメントのマップ。

注意 **sysaltusages** はローロック・カタログであるため、**reorg** を定期的を使用して、論理的に削除された領域を再利用する必要があります。

スクラッチ・データベースには、新しい **sysaltusages** テーブルが格納されません。スクラッチ・データベースは、**sysaltusages** テーブルを柔軟に配置するために使用されます。

どのデータベースでもスクラッチ・データベースに指定できます (master データベースやテンポラリ・データベースなどの例外を除く)。以下の理由から、スクラッチ・データベース専用のデータベースを使用することをおすすめします。

- **sysaltusages** のサイズが、サポートしているアーカイブ・データベースの数によって変化する場合があります。データベースのサイズを小さくすることはできないが、大きすぎる場合はそれを削除して、必要に応じて再作成できる。
- データベース・ログが自動的にトランケートされるように、"**trunc log on checkpoint**" オプションを有効にできる。

sysaltusages テーブルを持つ点を除けば、このデータベースは他のデータベースと同様です。スレッショルド・プロシージャと他の領域管理メカニズムを使用して、データベース内の領域を管理できます。

スクラッチ・データベース 次のように入力することにより、スクラッチ・データベースとして使用するデータベースを指定します。

```
sp_dboption <db name>, "scratch database", "true"
```

各アーカイブ・データベースに割り当てることができるスクラッチ・データベースは一度に 1 つのみですが、複数のデータベースが同じスクラッチ・データベースを使用することはできます。多数のアーカイブ・データベースがある場合は、複数のスクラッチ・データベースを定義できます。

sysattributes

すべてのデータベース

説明

システム属性は、データベース、テーブル、インデックス、ユーザ、ログイン、プロシージャなど、オブジェクトのプロパティを定義します。**sysattributes** には、(さまざまなシステム・プロシージャによって設定される)オブジェクトの属性定義ごとに 1 つのローがあります。**master.sysattributes** は、Adaptive Server 全体に対して、有効な属性値とクラスの設定を定義します。また、データベース、ログインなど、サーバワイドなオブジェクトの属性定義も格納します。

sysattributes へのアクセスには、システム・プロシージャのみを使用してください。**sysattributes** の変更に必要なパーミッションは、使用するシステム・プロシージャによって異なります。

カラム

sysattributes のカラムは次のとおりです。

名前	データ型	説明
class	smallint	属性のカテゴリを示す。 master.sysattributes では、特別クラス 0 が、Adaptive Server で有効なすべての属性のクラスを識別する。
attribute	smallint	属性 ID。 attribute は、タイプが EC でクラスが 25 であるオブジェクトについて、値が 1 (DECRYPT-DEFAULT_ID) の暗号化カラムに対するデフォルトの復号化方式を示す。
object_type	char(2)	1 文字または 2 文字の ID で、属性と関連付けるオブジェクトのタイプを定義する。
object_cinfo	varchar(255) null	オブジェクトに対する文字列の識別子 (たとえばアプリケーション名)。このフィールドは、すべての属性で使用されるわけではない。
object_cinfo2	varchar(255) null	SDC 環境でのオブジェクトに対する文字列の識別子 (たとえばアプリケーション名)。このフィールドは、すべての属性で使用されるわけではない。
object	int null	オブジェクト識別子。オブジェクトのタイプによって、オブジェクト ID、ユーザ ID、復号化デフォルト ID、データベース ID がある。オブジェクトがテーブルの一部 (たとえばインデックス) の場合、このカラムは関連するテーブルのオブジェクト ID を含む。
object_info1, object_info2, object_info3	int null	オブジェクトの識別に必要な追加情報を定義する。このフィールドは、すべての属性で使用されるわけではない。このフィールドの内容は定義する属性によって異なる。 <ul style="list-style-type: none"> • object_info_1 - 復号化デフォルト値を定義している暗号化カラムを持つテーブルのテーブル ID を表す。 • object_info2 - 復号化デフォルト値を格納する暗号化カラムの <i>colid</i> を示す。
int_value	int null	ユーザの表示レベルなどの属性に対する整数値。
char_value	varchar(768) null	キャッシュ名などの属性に対する文字値。
text_value	text null	属性に対するテキスト値。
image_value	image null	属性に対するイメージ値。
comments	varchar(255) null	属性の定義についてのコメントまたは追加情報。

表 1-1 には、`object_type` で最も頻繁に使用される値を示します。これらの値は、`sysattributes` に追加の情報を提供するためのものであり、単独で使用するものではありません。したがって、これらの値はクラス ID との組み合わせでのみ使用してください。

表 1-1: `sysattributes` の `object_type` カラムの有効な値

値	説明
D	データベース
I	インデックス
L	ログイン
P	プロシージャ
T	テーブル
U	ユーザ
AP	アプリケーション
DC	ダンプ条件
EL	外部ログイン (OMNI)
OD	オブジェクトの定義 (OMNI)
TC	トランザクション・コーディネーション (ASTC)
TG	テンポラリ・データベース・グループ (複数の tempdb)
TP	テキスト・ページ (OMNI)
QP	クエリ・プラン (抽象プラン)
UR	ユーザの役割
GR	グループの役割
LG	ログイン (MTDB バインド用)
EG	エンジン・グループ
PS	パスワードのセキュリティ

インデックス

- `class`, `attribute`, `object_type`, `object`, `object_info1`, `object_info2`, `object_info3`, `object_cinfo` カラムに、ユニーク・クラスタード・インデックスが作成されています。
- `object_type`, `object`, `object_info1`, `object_info2`, `object_info3`, `object_cinfo` カラムに、ノンクラスタード・インデックスが作成されています。

sysauditoptions

sybsecurity データベース

説明

sysauditoptions はサーバワイドな監査オプションごとに 1 つのローを持ち、そのオプションの現在の設定を表します。他の種類の監査オプションの設定は、他のテーブルに保存されます。たとえば、データベース固有のオプション設定は **sysdatabases** に保存され、オブジェクト固有のオプション設定は **sysobjects** に保存されます。各オプションのデフォルト値は 0 または “off” です。**sysauditoptions** にアクセスできるのはシステム・セキュリティ担当者だけです。

カラム

sysauditoptions のカラムは次のとおりです。

名前	データ型	説明
num	smallint	サーバワイドなオプションの数。
val	smallint	現在の値。次のうちのいずれか。 <ul style="list-style-type: none"> • 0 = off • 1 = pass • 2 = Failed • 3 = on
minval	smallint	このオプションに有効な最小値。
maxval	smallint	このオプションに有効な最大値。
name	varchar(30)	オプション名。
sval	varchar(30)	現在の値に相当する文字列。たとえば、“on”、“off”、“nonfatal”。
comment	varchar(255)	オプションの説明。

sysaudits_01 ~ sysaudits_08

syssecurity データベース

説明 これらのシステム・テーブルには監査証跡が含まれています。アクティブにできるテーブルは、一度に 1 つだけです。アクティブなテーブルは、**current audit table** 設定パラメータで決定されます。1 回のインストールで最大 8 つの監査テーブルを設定できます。たとえば、インストール環境に 3 つの監査テーブルがある場合、テーブル名は **sysaudits_01**、**sysaudits_02**、**sysaudits_03** になります。監査テーブルには、監査レコードごとに 1 つのローが含まれています。

カラム **sysaudits_01 ~ sysaudits_08** のカラムは次のとおりです。

名前	データ型	説明
event	smallint	監査されるイベントのタイプ。
eventmod	smallint	イベントに関する詳細情報。有効な値は次のとおりです。 <ul style="list-style-type: none"> 0 = このイベントの修飾子はない。 1 = イベントがパーミッションの検査に成功した。 2 = イベントがパーミッションの検査に失敗した。
spid	smallint int (Cluster Edition)	監査レコードを書き込むプロセスのサーバ・プロセス ID。
eventtime	datetime	監査イベントの日付と時刻。
sequence	smallint	1 つのイベントに対するレコードのシーケンス番号。イベントによっては複数の監査レコードが必要。
suid	smallint	監査イベントを実行したユーザのサーバ・ログイン ID。
dbid	int null	監査イベントが発生したデータベース ID。または、イベントのタイプによってオブジェクト、ストアド・プロシージャ、トリガのあるデータベース ID。
objid	int null	アクセスしたオブジェクト、またはストアド・プロシージャ、トリガの ID。
xactid	binary(6) null	監査イベントを含むトランザクション ID。マルチデータベース・トランザクションの場合は、トランザクションが発生したデータベースからのトランザクション ID。
loginname	varchar(30) null	suid に対応するログイン名。
dbname	varchar(30) null	dbid に対応するデータベース名。
objname	varchar(255) null	objid に対応するオブジェクト名。
objowner	varchar(30) null	objid の所有者名。
extrainfo	varchar(255) null	監査イベントについての追加情報。このフィールドの複数の項目は、セミコロンで区切られている。 表 1-2 を参照。
nodeid	tinyint null	今後のために予約済み (クラスタ環境では使用不可)。
instanceid	tinyint	インスタンスの ID (クラスタ環境のみで使用可能)。

注意 Cluster Edition ではデータ型にこの変更が加えられたため、監査テーブルをアップグレードする前にアーカイブおよびトランケートすることを強くおすすめします。それによって、**syssecurity** データベース内の領域不足によるアップグレードの失敗を低減できます。

`extrainfo` カラムにはセミコロンで区切られた項目列があります (表 1-2 を参照)。

表 1-2: `extrainfo` カラムの項目

項目	内容
Roles	アクティブな役割をリストする。役割はブランクで区切られている。
Keywords or options	イベントに使用されたキーワードまたはコマンド・オプションの名前。たとえば、 <code>alter table</code> コマンドでは、 <code>add column</code> または <code>drop constraint</code> オプションが使用される場合がある。複数のキーワードやオプションは、カンマで区切られている。
Previous value	イベントによって値が更新された場合には、更新される前の値。
Current value	イベントによって値が更新された場合には、新しい値。
Other information	イベントに記録されたセキュリティ関連のその他の情報。
Proxy information	<code>set proxy</code> が有効な間にイベントが発生した場合には、最初のログイン名。
Principal information	ユーザのログインがセキュア・デフォルト・ログインであり、ユーザが統一化ログインを使用して Adaptive Server にログインした場合に、基本となるセキュリティ・メカニズムのプリンシパル名が格納される。デフォルトのセキュア・ログインが使われない場合は、このフィールドの値は NULL。

監査設定パラメータを変更するセキュリティ関連イベントにおける `extrainfo` カラムの例を以下に示します。

```
sso_role;suspend auditing when full;1;0;;;
```

この例を見ると、システム・セキュリティ担当者が設定パラメータ `suspend auditing when full` を、1 (監査イベントのあるプロセスを中断) から 0 (次の監査テーブルをトランケートし、それを現在の監査テーブルにする) に変更したことがわかります。

syscharsets

master データベースのみ

説明 syscharsets には、Adaptive Server が使用する文字セットとソート順ごとに、1つのローが含まれています。ソート順の1つは、デフォルトのソート順として master..sysconfigures でマーク付けされており、これが実際に使用される唯一のソート順です。

カラム syscharsets のカラムは次のとおりです。

名前	データ型	説明
type	smallint	このローに表示されるエンティティのタイプ。1001 から 1999 までの番号は文字セットを表す。2000 から 2999 までの番号はソート順を表す。
id	tinyint	文字セットまたはソート順の ID。ソート順はソート順 ID と文字セット ID (csid) を組み合わせて定義される。文字セットはユニークな id によって定義される。0 ~ 200 までの ID 番号は Sybase によって予約されている。
csid	tinyint	ローが文字セットを表す場合、そのフィールドは使用されない。ローがソート順を表す場合は、ソート順が組み込まれる文字セットの ID。この ID を持つ文字セットのローは、このテーブルになければならない。
status	smallint	内部システム・ステータス情報ビット。
name	varchar(30)	文字セットまたはソート順のユニーク名。A ~ Z または a ~ z、0 ~ 9 の数字、およびアンダースコア (_) だけを含む 7 ビットの ASCII 文字で、文字で始まっているなければならない。
description	varchar(255)	文字セットまたはソート順の文字セット機能に関するオプションの説明。
definition	image	文字セットまたはソート順の内部定義。このフィールドのデータ構造は type によって異なる。
sortfile	varchar(30) null	ソート順ファイルの名前。

インデックス

- **id**、**csid** カラムに、ユニーク・クラスタード・インデックスが作成されています。
- **name** カラムに、ユニーク・ノンクラスタード・インデックスが作成されています。

syscolumns

すべてのデータベース

説明 syscolumns には、すべてのテーブルとビューのすべてのカラムに対して 1 つのロー、およびプロシージャのパラメータごとに 1 つのローが含まれます。

テーブルに関連する計算カラムおよび関数ベース・インデックス・キーそれぞれにつき 1 つのローが含まれます。

カラム syscolumns のカラムは次のとおりです。

名前	データ型	説明
id	int	このカラムが属するテーブル ID、またはこのパラメータに関連するプロシージャ ID。
number	smallint	プロシージャがグループ化されている場合のサブ・プロシージャ番号 (プロシージャ・エントリでない場合は 0)。
colid	smallint	カラム ID
status	tinyint	<ul style="list-style-type: none"> 0 ~ 2 ビット (値 1、2、4) - カラムが bit データ型を使用している場合には、ビットの位置を示す。カラムが text/image データ型を使用している場合には、ビット 0 と 1 は次のようにレプリケーション・ステータスを示す。 <ul style="list-style-type: none"> 01 = 常にレプリケート 10 = 変更された場合にのみレプリケート 00 = レプリケートしない ビット 3 (値 8) - NULL 値がこのカラムで有効かどうかを示す。 ビット 4 (値 16) - このカラムに複数の検査制約が存在するかどうかを示す。 ビット 5 と 6 - 内部使用。 ビット 7 (値 128) - IDENTITY カラムを示す。
type	tinyint	systypes からコピーされる物理記憶タイプ。
length	int	systypes からコピーされるか、またはユーザによって提供される物理データ長。
offset	smallint	このカラムを表示する場合のローのオフセット値。値が負の場合、そのカラムは可変長のカラムを示す。
usertype	smallint	systypes からコピーされるユーザ・タイプ ID。
cdefault	int	このカラムのデフォルト値を生成するプロシージャ ID。
domain	int	このカラムの最初のルールや検査制約の制約の ID。
name	varchar(255) not null	カラム名。
printfmt	varchar(255) null	予約済み。
prec	tinyint null	カラムのデータ型が numeric の場合の有効桁数。
scale	tinyint null	カラムのデータ型が numeric の場合の、小数点の右側の桁数。
remote_type	int null	リモート名にローカル名をマップする。この情報は、コンポーネント統合サービスのアクセス・メソッドにおいて、ソフトウェアがパラメータのネイティブなカラム・データ型情報を access_server クラスのサーバに渡すのに必要。
remote_name	varchar(255) null	リモート名にローカル名をマップする。この情報は、コンポーネント統合サービスのアクセス・メソッドにおいて、リモート・テーブルの適切なカラム名を使用してクエリを構築するのに必要。

名前	データ型	説明
xstatus	int null	拡張データ型を持つカラムのステータス。値は次のとおり。 <ul style="list-style-type: none"> 0 = in row 1 = off row xstatus は、拡張データ型を持たないカラムに対しては NULL となる。
xtype	int null	クラスの ID。 プロシージャのテーブルまたはパラメータ内のカラムがそのデータ型として Java クラスの場合に使用される。使用される場合、フィールドは NULL ではなく、type の値が 0x39 になる。詳細については、『Adaptive Server Enterprise における Java』を参照。
xdbid	int null	クラスのデータベース ID。それ以外の場合、値は現在のデータベース ID。 プロシージャのテーブルまたはパラメータ内のカラムがそのデータ型として Java クラスの場合に使用される。フィールドは NULL ではなく、type の値が 0x39 になる。詳細については、『Adaptive Server Enterprise における Java』を参照。
accessrule	int null	sysprocedures におけるアクセス・ルールのオブジェクト ID。詳細については、『システム管理ガイド』の「第 11 章 ユーザ・パーミッションの管理」の「ロー・レベル・アクセス制御」を参照。
status2	int null	SQLJ ストアド・プロシージャのパラメータ・モード、および SQLJ 関数の戻り型を示す。 計算カラムは次の 3 つの内部ビットがサポートしている。 <ul style="list-style-type: none"> 0x00000010、値 16 – カラムは計算カラム。 0x00000020、値 32 – カラムはマテリアライズされた計算カラム。 0x00000040、値 64 – カラムはビュー内の計算カラム。 0x00001000、値 4096 – 復号化デフォルトがある暗号化カラム。 syscolumns の status2 フィールドは、次のコード化を使用してカラムの暗号化プロパティを示す。 <ul style="list-style-type: none"> 0x80、値 128 – カラムは暗号化済み。 0x100、値 256 – カラムは、初期化ベクトルを使用して暗号化済み。 0x200、値 512 – カラムは、ランダム埋め込みを使用して暗号化済み。 0x400、値 1024 – プロキシ・テーブルは暗号化済み。 0x1000、値 4096 – 復号化デフォルトがある暗号化カラム。
status3	int	0x0001、値 1 – 関数ベース・インデックス・キーに対する隠し計算カラムを示す。
computedcol	int	計算カラム定義のオブジェクト ID を格納する。
encrtype	int null	暗号化された状態でのデータ型。
enclen	int null	暗号化データの長さ。
encykeyid	int null	キーのオブジェクト ID。
encykeydb	varchar(30) null	暗号化キーが作成されたデータベースの名前。暗号化カラムと同じデータベースの場合は NULL。
enccrdate	datetime null	暗号化キーの作成日。sysobjects.crdate からコピーされる。

インデックス

- id、number、colid カラムに、ユニーク・クラスタード・インデックスが作成されています。

syscomments

すべてのデータベース

説明

syscomments テーブルには、ビュー、ルール、デフォルト、トリガ、テーブル制約、パーティション、プロシージャ、計算カラム、関数ベース・インデックス・キー、その他の形式のコンパイル済みオブジェクトのそれぞれに対するエントリが含まれています。**text** カラムには、実際の定義文が含まれています。**text** カラムが 255 バイトより長い場合、エントリは複数のローにわたって生成されます。それぞれのオブジェクトは最大 65,025 ローまで使用できます。

また、計算カラム、関数ベース・インデックス、またはパーティション定義のテキストも格納されます。たとえば、範囲パーティションの場合は“*values <= value_list*”です。

create service コマンドでは、**create procedure** インフラストラクチャが使用され、**syscomments** にテキストが格納されます。

カラム

syscomments のカラムは次のとおりです。

名前	データ型	説明
id	int	このテキストが適用されるオブジェクト ID。
number	smallint	プロシージャがグループ化されている場合のサブ・プロシージャ番号 (プロシージャ・エントリでない場合は 0)。
colid	smallint	このプロシージャのコメントのカラム・カウンタの下位部分。範囲は 0 ~ 32767。プロシージャのテキストがその数のローに収まらない場合、このカウンタは colid2 と組み合わせて機能する。
texttype	smallint	コメントのタイプを示す。値は次のとおり。 <ul style="list-style-type: none"> 0 - ビュー、ルール、デフォルト、トリガ、プロシージャについてシステムが提供するコメント。 1 - オブジェクトまたはカラムを説明するエントリを追加するためにユーザが指定するコメント。
language	smallint	予約済み。
text	varchar(255) null	SQL 定義文の実際のテキストを示す。
colid2	smallint	このプロシージャのコメントのカラム・カウンタの上位部分。範囲は 0 ~ 32767。この値が 1 以上になるのは、プロシージャ・テキストのローの数が 32,768 より多いプロシージャの場合のみ。
status	smallint null	オブジェクトのステータスを示すビット。 <ul style="list-style-type: none"> 0x1 - SYSCOM_TEXT_HIDDEN はテキストが隠されていることを示す。 0x2 - 内部使用のために予約済み。 0x4 - SYSCOM_QUOTED_ID_ON はオブジェクト作成時に引用符で囲まれた識別子がオンになっていたことを示す。
partitionid	int null	それ以外は null。

注意 定義文は、**syscomments** のテキスト・カラムから削除しないでください。定義文を暗号化するには、システム・プロシージャ **sp_hidetext** を実行します。バージョン 11.5 以降で作られた文が削除されたかどうかを確認するには、**sp_checksourc**e を実行します。

データベース・オブジェクトのテキストに対する不正なアクセスを防ぐには、**syscomments** テーブルの **text** カラム上の **select** パーミッションを、所有者とシステム管理者に限定します。この制限は、ストアド・プロシージャによるアクセスだけでなく、**select** 文を使った直接アクセスにも適用され、評価済み設定で Adaptive Server を実行するときが必要です。この制限を有効にするには、システム・セキュリティ担当者が **allow select on syscomments.text column** というパラメータをシステム・プロシージャ **sp_configure** で再設定する必要があります。評価済み設定の詳細については、『システム管理ガイド』を参照してください。

インデックス

- **id**、**number**、**colid2**、**colid**、**texttype** カラムに、ユニーク・クラスタード・インデックスが作成されています。

sysconfigures

master データベースのみ

説明 sysconfigures には、設定パラメータごとに 1 行のローが含まれています。これらはユーザが設定できます。

カラム sysconfigures のカラムは次のとおりです。

名前	データ型	説明
config	smallint	設定パラメータの番号。
value	int	integer データ型によってユーザが変更できるパラメータの値。character データ型のパラメータでは、値は 0。
comment	varchar(255)	設定パラメータの名前。
status	int	設定パラメータの型を表す値。詳細については、表 1-3 を参照。
name	varchar(255) null	設定パラメータの名前 (comment と同じ値)。
parent	smallint null	親の設定パラメータの数。複数の親が存在する場合、追加の親の数は sysattributes に保管される。
value2	varchar(255) null	character データ型によってユーザが変更できるパラメータの値。integer データ型のパラメータの場合、値は NULL。value2 は、バッファ・プールのプール・サイズを格納するためにも使用される。
value3	int null	バッファ・プールのウォッシュ・サイズを保管する。
value4	int null	バッファ・プールの非同期プリフェッチ率を保管する。指定なしまたはデフォルト値の場合は -1。
instanceid	tinyint	インスタンスの ID。クラスタ環境のみで使用可能。

表 1-3 に、status カラムに関する情報を示します。

表 1-3: status カラムの説明

ステータス・タイプ	10 進数	16 進数	説明
CFG_NO_OPTIONS	0	0x0	パラメータ・オプションなし
CFG_SYSTEM_OPTION	1	0x01	パラメータとしてシステム・オプションを指定
CFG_SYSTEM_GROUP	2	0x02	パラメータとしてシステム・グループを指定
CFG_STATIC	4	0x04	静的パラメータを指定
CFG_DYNAMIC	8	0x08	動的パラメータを指定
CFG_CALCULATED	16	0x10	算出パラメータを指定
CFG_READONLY	32	0x20	読み取り専用パラメータを指定
CFG_MEMORY_USED	64	0x40	パラメータ設定によるメモリ消費が発生
CFG_CONFIG_FILE	128	0x80	外部開示パラメータ
CFG_SYSTEM_TAB	256	0x100	システム・テーブルでのみパラメータを外部開示
CFG_EXTRAS_OPTION	512	0x200	CFG_EXTRAS 用パラメータ (DS_CONFIG 用ではない)
CFG_CFGBLK	1024	0x400	パラメータは設定ブロックに格納
CFG_CACHE_GROUP	2048	0x800	パラメータとしてキャッシュ・グループを指定
CFG_CACHE_OPTION	4096	0x1000	パラメータとしてキャッシュ・オプションを指定

ステータス・タイプ	10 進数	16 進数	説明
CFG_BUFFER_POOL_GROUP	8192	0x2000	パラメータとしてバッファ・プール・グループを指定
CFG_BUFFER_POOL_OPTION	16384	0x4000	パラメータとしてバッファ・プール・オプションを指定
CFG_INTERNAL	32768	0x8000	内部使用専用のパラメータ
CFG_FNOF_LPAGESIZE	65536	0x10000	パラメータ・エントリは論理ページサイズに依存

- インデックス
- name、parent、config カラムに、ユニーク・クラスタード・インデックスが作成されています。
 - config カラムに、ノンクラスタード・インデックスが作成されています。
 - parent、config カラムに、ノンクラスタード・インデックスが作成されています。

sysconstraints

すべてのデータベース

説明

`create table` または `alter table` を使用して、新しい検査制約または参照制約を宣言すると、Adaptive Server によってローが `sysconstraints` テーブルに挿入されます。`alter table` を実行して制約を削除するまで、ローはそのまま残ります。`drop table` を実行してテーブルを削除すると、`sysconstraints` テーブルからそのテーブルに関連するすべてのローが削除されます。

このテーブルには、特定のテーブルに関連する検査制約、参照制約、計算カラム、および関数ベース・インデックス・キーごとに 1 つのローが含まれています。

カラム

`sysconstraints` のカラムは次のとおりです。

名前	データ型	説明
<code>colid</code>	<code>smallint</code>	テーブル内のカラム番号。
<code>constrid</code>	<code>int</code>	制約のオブジェクト ID。
<code>tableid</code>	<code>int</code>	制約が宣言されるテーブルの ID。
<code>error</code>	<code>int</code>	制約ごとのエラー・メッセージ。
<code>status</code>	<code>int</code>	制約のタイプ： <ul style="list-style-type: none"> • 0x0040 = 参照制約 • 0x0080 = 検査制約 • 0x0100 = 計算カラム・オブジェクト制約
<code>spare2</code>	<code>int</code>	未使用。

インデックス

- `tableid`、`colid` カラムに、ユニーク・クラスタード・インデックスが作成されています。
- `constrid` カラムに、ノンクラスタード・インデックスが作成されています。

syscoordinations

sysystemdb データベースのみ

説明 syscoordinations には、分散トランザクション (リモート・パティシパント) に関与するリモート Adaptive Server とそれらのコーディネーション・ステータスについての情報が含まれています。

カラム syscoordinations のカラムは次のとおりです。

名前	データ型	説明
participant	smallint	パティシパント ID。
starttime	datetime	トランザクションが開始した日付。
coordtype	tinyint	sysransactions テーブル定義内のコーディネーションの方法かプロトコルを示す値。
owner	tinyint	ローの所有者 (内部使用) 。
protocol	smallint	内部使用。
state	int	リモート・パティシパントの現在のステータスを示す値 (表 1-4 を参照) 。
bootcount	int	内部使用。
dbid	smallint	トランザクションの開始時のデータベース ID。
logvers	tinyint	内部使用。
spare	tinyint	内部使用。
status	int	内部使用。
xactkey	binary(14)	ユニーク Adaptive Server トランザクション・キー。
gtrid	varchar(255) null	Adaptive Server によってコーディネートされた分散トランザクションのグローバル・トランザクション ID (内部使用) 。
partdata	varbinary(255) null	内部使用。
srvname	varchar(30) null	ローカル・サーバ用の名前 (リモート・サーバの場合は null) 。
nodeid	tinyint null	クラスター環境以外では使用不可 - 今後のために予約済み。
instanceid	tinyint	クラスター環境専用 - インスタンスの ID。

表 1-4 に state カラムの値をリストします。

表 1-4: syscoordinations のステータス値

state 値	パティシパント・ステータス
1	Begun
4	Prepared
7	Committed
9	In Abort Tran

インデックス

- xactkey、participant、owner カラムに、ユニーク・クラスタード・インデックスが作成されています。

syscurconfigs

master データベースのみ

説明 `syscurconfigs` テーブルはクエリを実行するときに動的に構築されます。これには `sysconfigures` と同じように各設定パラメータのエントリが含まれていますが、その値はデフォルト値ではなく現在の値です。さらに、設定構造を説明する 4 つのローが含まれています。

カラム `syscurconfigs` のカラムは次のとおりです。

名前	データ型	説明
config	smallint	設定パラメータの番号。
value	int	<code>integer</code> データ型のパラメータの現在の実行値。 <code>character</code> データ型のパラメータでは、値は 0。
comment	varchar(255)	設定パラメータについてのコメント。内部用。
status	int	設定パラメータの型を表す値。詳細については、表 1-5 を参照。
value2	varchar(255) null	<code>character</code> データ型のパラメータの現在の実行値を示す。 <code>integer</code> データ型のパラメータの場合、値は NULL。
defvalue	varchar(255) null	設定パラメータのデフォルト値。
minimum_value	int null	設定パラメータの最小値。
maximum_value	int null	設定パラメータの最大値。
memory_used	int null	各設定パラメータが使用するメモリ量の整数値。負の値はメモリ共有を示す。
display_level	int null	設定パラメータの表示レベル。値は 1、5、10。
datatype	int null	設定パラメータのデータ型。
message_num	int null	この設定パラメータ用の <code>sp_helpconfig</code> メッセージのメッセージ番号。
apf_percent	int null	バッファ・プールの非同期ブリフェッチ率の現在の実行値。バッファ・プールを示すローでのみ有効。
nodeid	tinyint null	今後のために予約済み (クラスタ環境では使用不可)。
instanceid	tinyint	インスタンスの ID (クラスタ環境のみで使用可能)。 <ul style="list-style-type: none"> • Not applicable – パラメータに単位はない • Kilobytes • Number – 項目数 • Megabytes • Clock ticks – クロック・チック数 • Memory pages (2K) • Microseconds • Virtual pages (2K) • Milliseconds • Logical pages • Seconds • Percent • Minutes • Ratio • Hours • Switch – プール値 • Days • ID – ID 番号 • Bytes • Name • Rows

名前	データ型	説明
type	varchar(10) null	設定パラメータがその構造体の定義で動的と宣言されるか、静的と宣言されるかを示す。値は次のとおり。 <ul style="list-style-type: none"> • Dynamic – すぐに有効になる • Static – Adaptive Server の再起動後に有効になる

表 1-5 に、status カラムに関する情報を示します。

表 1-5: status カラムの説明

ステータス・タイプ	10 進数	16 進数	説明
CFG_NO_OPTIONS	0	0x0	パラメータ・オプションなし。
CFG_SYSTEM_OPTION	1	0x01	パラメータとしてシステム・オプションを指定。
CFG_SYSTEM_GROUP	2	0x02	パラメータとしてシステム・グループを指定。
CFG_STATIC	4	0x04	静的パラメータを指定。
CFG_DYNAMIC	8	0x08	動的パラメータを指定。
CFG_CALCULATED	16	0x10	算出パラメータを指定。
CFG_READONLY	32	0x20	読み取り専用パラメータを指定。
CFG_MEMORY_USED	64	0x40	パラメータ設定によるメモリ消費が発生。
CFG_CONFIG_FILE	128	0x80	外部開示パラメータ。
CFG_SYSTEM_TAB	256	0x100	システム・テーブルでのみパラメータを外部開示。
CFG_EXTRAS_OPTION	512	0x200	CFG_EXTRAS 用パラメータ (DS_CONFIG 用ではない)。
CFG_CFGBLK	1024	0x400	パラメータは設定ブロックに格納。
CFG_CACHE_GROUP	2048	0x800	パラメータとしてキャッシュ・グループを指定。
CFG_CACHE_OPTION	4096	0x1000	パラメータとしてキャッシュ・オプションを指定。
CFG_BUFFER_POOL_GROUP	8192	0x2000	パラメータとしてバッファ・プール・グループを指定。
CFG_BUFFER_POOL_OPTION	16384	0x4000	パラメータとしてバッファ・プール・オプションを指定。
CFG_INTERNAL	32768	0x8000	内部使用専用のパラメータ。
CFG_FNOF_LPAGESIZE	65536	0x10000	パラメータ・エントリは論理ページサイズに依存。

sysdatabases

master データベースのみ

説明 `sysdatabases` テーブルには、Adaptive Server のデータベースごとに 1 つのローが含まれます。Adaptive Server をインストールすると、`sysdatabases` には、`master` データベース、`model` データベース、`sysystemprocs` データベース、および `tempdb` データベースのエントリが含まれます。監査をインストールした場合は、`syssecurity` データベースのエントリも含まれます。

カラム `sysdatabases` のカラムは次のとおりです。

名前	データ型	説明
<code>name</code>	<code>sysname</code>	データベースの名前。
<code>dbid</code>	<code>smallint</code>	データベース ID。
<code>suid</code>	<code>int</code>	データベース所有者のサーバ・ユーザ ID。
<code>status</code>	<code>smallint</code>	ユーザが <code>sp_dboption</code> で設定できる制御ビット。詳細については、 表 1-6 を参照。
<code>version</code>	<code>smallint</code>	未使用。
<code>logptr</code>	<code>int</code>	トランザクション・ログへのポインタ。
<code>crdate</code>	<code>datetime</code>	作成した日付。
<code>dumptrdate</code>	<code>datetime</code>	最新の <code>dump transaction</code> の日付。
<code>status2</code>	<code>smallint null</code>	追加の制御ビット (表 1-7 (30 ページ) を参照)。
<code>audflags</code>	<code>int null</code>	データベースの監査設定。
<code>deftabaud</code>	<code>int null</code>	テーブルのデフォルト監査設定を定義するビットマスク。
<code>defvwaud</code>	<code>int null</code>	ビューのデフォルト監査設定を定義するビットマスク。
<code>defpraud</code>	<code>int null</code>	ストアド・プロシージャのデフォルト監査設定を定義するビットマスク。
<code>def_remote_type</code>	<code>smallint null</code>	ストアド・プロシージャ <code>sp_addobjectdef</code> によって記憶位置が提供されていない場合には、リモート・テーブルで使用するデフォルトのオブジェクト・タイプを示す。
<code>def_remote_loc</code>	<code>varchar(349) null</code>	ストアド・プロシージャ <code>sp_addobjectdef</code> によって記憶位置が提供されていない場合には、リモート・テーブルで使用するデフォルトの記憶位置を示す。
<code>status3</code>	<code>int null</code>	追加の制御ビット。
<code>status4</code>	<code>int null</code>	追加の制御ビット。
<code>audflags2</code>	<code>varbinary(16) null</code>	今後のために予約済み。
<code>instanceid</code>	<code>tinyint</code>	インスタンスの ID (Cluster Edition のみ)。
<code>durability</code>	<code>int</code>	データベースの持続性レベル。値は次のとおり。 1 – full 5 – at_shutdown 6 – no_recovery

表 1-6 に、status カラムのビット表示をリストします。

表 1-6: sysdatabases テーブルのステータス制御ビット

10 進数	16 進数	ステータス
1	0x01	このデータベースからアップグレードが開始された。
2	0x02	アップグレードに成功した。
4	0x04	<ul style="list-style-type: none"> select into/bulkcopy ユーザによって設定可能。
8	0x08	<ul style="list-style-type: none"> trunc log on chkpt ユーザによって設定可能。
16	0x10	<ul style="list-style-type: none"> no chkpt on recovery ユーザによって設定可能。
32	0x20	for load オプションを使用してデータベースが作成されたか、データベースをロードしているときにクラッシュして、リカバリされていない。
64	0x04	リカバリするすべてのデータベースのリカバリが開始された。
256	0x100	<ul style="list-style-type: none"> データベースが疑わしい (suspect 状態)。 リカバリされていない。 データベースをオープンまたは使用することができない。 dbcc dbrepair でのみ削除可能。
512	0x200	<ul style="list-style-type: none"> ddl in tran ユーザによって設定可能。
1024	0x400	<ul style="list-style-type: none"> read only ユーザによって設定可能。
2048	0x800	<ul style="list-style-type: none"> dbo use only ユーザによって設定可能。
4096	0x1000	<ul style="list-style-type: none"> single user ユーザによって設定可能。
8192	0x2000	<ul style="list-style-type: none"> allow nulls by default ユーザによって設定可能。

表 1-7 に、status2 カラムのビット表示をリストします。

表 1-7: sysdatabases テーブルの status2 制御ビット

10 進数	16 進数	ステータス
1	0x0001	abort tran on log ful (ユーザによって設定可能)。
2	0x0002	no free space acctg (ユーザによって設定可能)。
4	0x0004	auto identity (ユーザによって設定可能)。
8	0x0008	identity in nonunique index (ユーザによって設定可能)。
16	0x0010	データベースがオフライン状態になっている。
32	0x0020	リカバリが完了するまでデータベースがオフライン状態になっている。
64	0x0040	テーブルには auto identity 機能があり、identity カラムに一意性制約がある。
128	0x0080	データベースに疑わしいページがある。

10 進数	16 進数	ステータス
256	0x0100	テーブル構造がディスクに書き込まれる。リカバリの完了後にこのビットが表示された場合、オープン・データベースに対してサーバの設定が不足している可能性がある。このパラメータを増やすには、 <code>sp_configure</code> を使用する。
512	0x0200	データベースはアップグレード処理中。
1024	0x0400	データベースは、スタンバイ・アクセス用にオンライン状態。
2048	0x0800	ユーザによって設定されると、エイリアスを使用したデータベース間アクセスを防止する。
-32768	0xFFFF8000	データベースにログ専用のデバイスではないログの一部がある。

表 1-8 に、`status3` カラムのビット表示をリストします。

表 1-8: `sysdatabases` テーブルの `status3` 制御ビット

10 進数	16 進数	ステータス
0	0x0000	通常のデータベースまたは標準のデータベース、あるいは <code>create</code> 文でプロキシ更新のないデータベース。
1	0x0001	<code>proxy_update</code> オプションが指定されており、データベースはユーザが作成したプロキシ・データベースである。
2	0x0002	データベースは高可用性によって作成されたプロキシ・データベースである。
4	0x0004	データベースに、高可用性によって作成されたプロキシ・データベースがある。
8	0x0008	データベースが停止されるため、データベースのアクセスを禁止する。
16	0x0010	データベースはフェールオーバ・データベースである。
32	0x0020	データベースは <code>master</code> の型のマウントされたデータベースである。
64	0x0040	データベースはマウントされたデータベースである。
128	0x0080	<code>quiesce database</code> コマンドによってデータベースへの書き込みがブロックされる。
256	0x0100	ユーザ作成の <code>tempdb</code> 。
512	0x0200	フェールオーバ状態のサーバのデータベースへの外部アクセスを禁止する。
1024	0x0400	非同期ロギング・サービス・スレッドを有効または無効にするための、ユーザ提供のオプション。ユーザは、特定のデータベースで <code>sp_dboption enable async logging service</code> オプションを <code>true</code> に設定してこれを有効にする。
4096	0x1000	データベースが正常に停止された。
8192	0x2000	<code>drop database</code> が進行中である。

インデックス

- `name` カラムに、ユニーク・クラスタード・インデックスが作成されています。
- `dbid` カラムに、ノンクラスタード・インデックスが作成されています。

sysdepends

すべてのデータベース

説明 sysdepends には、プロシージャ、ビュー、またはトリガによって参照される各プロシージャ、ビュー、またはテーブル用にローが 1 つずつ含まれています。

カラム sysdepends のカラムは次のとおりです。

名前	データ型	説明
id	int	オブジェクト ID。
number	smallint	プロシージャ数。
depid	int	従属オブジェクト ID。
depnumber	smallint	従属プロシージャ数。
status	smallint	内部ステータス情報。
selall	bit	オブジェクトが <code>select * 文</code> で使用される場合は on。
resultobj	bit	オブジェクトが更新されている場合は on。
readobj	bit	オブジェクトが読み込まれている場合は on。
columns	varbinary	ストアド・プロシージャの本体で参照されているカラムのカラム ID のビットマップを格納する。このビットマップからは、コンパイル済みオブジェクトについて、カラムレベルの依存性追跡情報が得られ、 <code>sp_depends</code> によって解析されて、ストアド・プロシージャ、トリガ、ビューのカラムレベルの依存性がレポートされる。

インデックス

- `id`、`number`、`depid`、`depnumber` カラムに、ユニーク・クラスタード・インデックスが作成されています。

sysdevices

master データベースのみ

説明

sysdevices には、テープ・ダンプ・デバイス、ディスク・ダンプ・デバイス、データベース用ディスク、およびデータベース用ディスク・パーティションそれぞれにローが1つずつ含まれています。Adaptive Server の配布メディアの **sysdevices** には、マスタ・デバイス用 (データベース用) に1つ、ディスク・ダンプ・デバイス用に1つ、テープ・ダンプ・デバイス用に2つ、合計4つのエントリがあります。

注意 Adaptive Server バージョン 15.0 では、デバイスの ID 番号は **vdevno** カラムに格納されており、**high** カラムや **low** カラムには含まれていません。したがって、以前のスキーマに基づいてデバイス ID 番号を判断するスクリプトとストアド・プロシージャは、修正が必要な場合があります。

カラム

sysdevices のカラムは次のとおりです。

名前	データ型	説明
low	int	ダンプ・デバイスには使用しない - 2K バイトの仮想ページのブロック・オフセット。
high	int	2K バイトの最後の仮想ページのブロック・オフセット
status	smallint	デバイス・タイプ、デフォルト・ステータス、およびミラー・ステータスを示すビット・マップ (表 1-9 を参照)。
cntrtype	smallint	コントローラ・タイプ。 <ul style="list-style-type: none"> • 0 = データベース・デバイス • 2 = ディスク・ダンプ・デバイスまたはストリーム・テープ • 3 ~ 8 = テープ・ダンプ・デバイス
name	sysname	ダンプ・デバイス、データベース・デバイス、またはメモリ内記憶域キャッシュの論理名。
phyname	varchar(127)	物理デバイスまたはメモリ内記憶域キャッシュの名前。
mirrorname	varchar(127) null	ミラー・デバイス名。
vdevno	int	デバイス ID 番号。
crdate	datetime null	デバイスが追加された日付。
resizedate	datetime null	このデバイスに対して disk resize が最後に実行された日付。
status2	int	このデバイスの追加のステータス・ビット (表 1-10 を参照)。
instanceid	tinyint	インスタンスの ID (クラスタ環境のみで使用可能)。
uuid	varbinary(16)	今後のために予約済み (クラスタ環境のみで使用可能)。

status カラムのビット表示は、以下のように追加できます。たとえば、“3” は物理ディスクであると同時に、デフォルトであることを示します。

表 1-9: status カラムのビット表示

10 進数	16 進数	ステータス
1	0x01	デフォルトのディスク。
2	0x02	物理ディスク。
4	0x04	未使用 - 論理ディスク。
8	0x08	ヘッダが省略されている。
16	0x10	ダンプ・デバイス。
32	0x20	逐次書き込み。
64	0x40	デバイスがミラーリングされている。
128	0x80	読み込みがミラーリングされている。
256	0x100	セカンダリ・ミラーリング側のみ。
512	0x200	ミラーリング使用可能。
1024	0x400	マスタ・デバイスがミラーリングされている。
2048	0x800	内部使用 - ミラーリングが無効である。
4096	0x1000	内部使用 - プライマリ・デバイスのミラーリング解除が必要。
8192	0x2000	内部使用 - セカンダリ・デバイスのミラーリング解除が必要。
16384	0x4000	UNIX ファイル・デバイスは、 <code>dsync</code> 設定を使用する (物理メディアに直接書き込みが行われる)。

表 1-10 に、`status2` カラムのビット表示を示します。

表 1-10: status2 カラムのビット表示

10 進数	16 進数	ステータス
1	0x01	このデバイスではダイレクト I/O が有効になっている。

インデックス

- `name` カラムに、ユニーク・クラスタード・インデックスが作成されています。

sysencryptkeys

すべてのデータベース

説明 デフォルト・キーも含め、データベースに作成される各キーには、データベース固有のシステム・カタログ `sysencryptkeys` にエントリがあります。

カラム `sysencryptkeys` のカラムは次のとおりです。

カラム名	データ型	説明
id	int	暗号化キー ID。
ekalgorithm	int	暗号化アルゴリズム。
type	smallint	キーの種類。値は次のとおり。 <ul style="list-style-type: none"> 0x1 (10 進数 1) – 対称キー 0x4 (10 進数 4) – デフォルト・キー 0x10 (10 進数 16) – キー・コピー 0x40 (10 進数 64) – リカバリ・キー・コピー
status	int	内部ステータス情報。ビット表示は次のとおり。 <ul style="list-style-type: none"> 0x1 (10 進数 1) – キーは初期化ベクトルを使用 0x2 (10 進数 2) – キーはランダム埋め込みを使用 0x4 (10 進数 4) – キーは紛失したパスワードを保護するために暗号化済み 0x8 (10 進数 8) – キー・コピーはログイン・アクセスのために暗号化済み 0x10 (10 進数 16) – キー・コピーはログイン・パスワードで暗号化済み 0x20 (10 進数 32) – キー・コピーはシステム暗号化パスワードで暗号化済み 0x100 (10 進数 256) – キーはユーザ・パスワードで暗号化済み
eklen	smallint	ユーザが指定したキーの長さ。
value	varbinary(1282)	キーの暗号値。該当するキーの対称暗号化が示される。キーを暗号化するために、Adaptive Server は AES でシステム暗号化パスワード、ユーザ指定パスワード、またはログイン・パスワードの 128 ビット・キーを使用する。
uid	int null	キー・コピーが割り当てられているユーザのユーザ ID。
eksalt	varbinary(20)	暗号化キーの復号化の検証に使用する乱数値。
ekpairid	int null	未使用。
pwdate	datetime null	最後にパスワードを変更した日付。
expdate	int null	未使用。
ekpwdwarn	int null	未使用。

sysencryptkeys のステータス・ビット。

表 1-11: sysencryptkeys のステータス・ビット

10 進数	16 進数	ステータス
	0x00000004	EK_KEYRECOVERY() – 紛失したパスワードを保護するために暗号化されたキー
	0x00000008	EK_LOGINACCESS() – ログイン・アクセスのために暗号化されたキー
	0x00000010	EK_LOGINPASS () – ログイン・パスワードを使用して暗号化されたキー
	0x00000100	EK_USERPWD() – ユーザ暗号化パスワードを使用して暗号化されたキー

sysengines

master データベースのみ

説明 sysengines には、現在オンラインになっている Adaptive Server エンジンごとに 1 つのローが格納されます。

カラム sysengines のカラムは次のとおりです。

名前	データ型	説明
engine	smallint	エンジン番号。
osprocid	int	<ul style="list-style-type: none"> UNIXプラットフォーム – オペレーティング・システムのプロセス ID (NULL の場合もある)。 Windows – スレッド・ハンドル。
osprocname	char(32)	オペレーティング・システムのプロセス名 (NULL の場合もある)。
status	char(12)	オンライン、オフライン、作成中、削除中、デバッグ、不良のいずれかのステータスを示す。
affinitied	int	エンジンと結び付いている Adaptive Server プロセス番号。
cur_kpid	int	エンジンで現在実行中のプロセスがある場合のカーネル・プロセス ID。
last_kpid	int	エンジンで以前に実行したプロセスのカーネル・プロセス ID。
idle_1	tinyint	予約済み。
idle_2	tinyint	予約済み。
idle_3	tinyint	予約済み。
idle_4	tinyint	予約済み。
starttime	datetime	エンジンがオンラインになった日付と時刻。
nodeid	tinyint null	今後のために予約済み (クラスタ環境では使用不可)。
instanceid	tinyint	インスタンスの ID (クラスタ環境のみで使用可能)。

sysgams

すべてのデータベース

説明

sysgams には、データベースのグローバル・アロケーション・マップ (GAM) が保存されます。GAM は、アロケーション・ユニット当たり 1 ビットで、データベースの全アロケーション・ユニットのビットマップを保存します。**sysgams** から選択したり、参照したりすることはできません。

sysindexes

すべてのデータベース

説明

sysindexes には、クラスタード・インデックスごとに 1 つのロー、ノンクラスタード・インデックスごとに 1 つのロー、クラスタード・インデックスのないテーブルごとに 1 つのロー、**text** カラムまたは **image** カラムのあるテーブルごとに 1 つのローが含まれています。このテーブルには、関数ベース・インデックスや計算カラムに作成されたインデックスごとにも 1 つのローが含まれています。

カラム

sysindexes のカラムは次のとおりです。

名前	データ型	説明
name	varchar(255) null	インデックスまたはテーブル名。
id	int	インデックスの ID、またはインデックスが設定されたテーブルの ID。
indid	smallint	有効な値は次のとおり。 <ul style="list-style-type: none"> • 0 = テーブル • 1 = 全ページロック・テーブルのクラスタード・インデックス • 2 以上 = ノンクラスタード・インデックスまたはデータオンリーロック・テーブルのクラスタード・インデックス • 255 = text、image、テキスト・チェーン、または Java の off-row 構造 (ラージ・オブジェクト (LOB) 構造)
doampg	int	旧式。
ioampg	int	旧式。
oampgtrips	int	キャッシュ内で再使用されない状態で OAM ページがフラッシュされるまでのサイクル数。
status3	smallint	内部システム・ステータス情報。
status2	smallint	内部システム・ステータス情報 (表 1-13 を参照)。
ipgtrips	int	キャッシュ内で再使用されない状態でインデックス・ページがフラッシュされるまでのサイクル数。
first	int	旧式。
root	int	旧式。
distribution	int	未使用。以前は、インデックスの分散ページのページ番号を格納するために使用した。
usagecnt	smallint	予約済み。
segment	smallint	オブジェクトがあるセグメントの番号。
status	smallint	内部システム・ステータス情報 (表 1-12 を参照)。
maxrowsperpage	smallint	1 ページあたりの最大ロー数。
minlen	smallint	ローの最小サイズ。
maxlen	smallint	ローの最大サイズ。
maxirow	smallint	リーフではないインデックスのローの最大サイズ。

名前	データ型	説明
keycnt	smallint	全ページ・ロック・テーブル上のクラスタード・インデックスのキーの数。クラスタード・インデックスの場合はキーの数で、それ以外のインデックスの場合はキーの数+1。
keys1	varbinary(255) null	キー・カラムの説明 (エントリがインデックスの場合)。
keys2	varbinary(255) null	キー・カラムの説明 (エントリがインデックスの場合)。
soid	tinyint	インデックスを作成したときのソート順の ID。キーに文字データがない場合は 0。
csid	tinyint	インデックスを作成したときの文字セットの ID。キーに文字データがない場合は 0。
base_partition	int null	旧式。
fill_factor	smallint null	sp_chgattribute を使用して設定したテーブルの fillfactor の値。
res_page_gap	smallint null	テーブルの reservepagegap の値。
exp_rowsize	smallint null	予期されるデータ・ロー・サイズ。
keys3	varbinary(255) null	キー・カラムの説明 (エントリがインデックスの場合)。
identitygap	int null	テーブルのアイデンティティ・ギャップ。
crdate	datetime null	作成した日付。
partitiontype	smallint null	値は次のとおり。 <ul style="list-style-type: none"> • 1 = 範囲 • 2 = ハッシュ • 3 または NULL = [デフォルト] ラウンドロビン • 4 = リスト
conditionid	int null	パーティション条件の ID。partitiontype がラウンドロビンまたはハッシュの場合は null。

表 1-12 に、status カラムのビット表示をリストします。

表 1-12: sysindexes テーブルの status カラムのステータス・ビット

10 進数	16 進数	ステータス
1	0x1	重複キーを挿入しようとした場合に、現在のコマンドまたはトリガをアボートする。
2	0x2	ユニーク・インデックス。
4	0x4	重複ローを挿入しようとした場合、現在のコマンドまたはトリガをアボートする。データオンリーロック・テーブルの場合、常に 0。
16	0x10	クラスタード・インデックスがある全ページロック・テーブル。
64	0x40	全ページ・ロック・テーブルの場合、インデックスに重複ローを使用できる。データオンリーロック・テーブルの場合、常に 0。
128	0x80	内部使用されているソート済みオブジェクトの切り替え。create clustered index、reorg rebuild、または alter table ロック・スキーム・コマンドを使用して設定する。

10 進数	16 進数	ステータス
512	0x200	create index 文で sorted data オプションが使用されている。
2048	0x800	プライマリ・キー上のインデックスである。
32768	0x8000	疑わしい (suspect 状態) インデックス。インデックスは、別のソート順を使用して作成されている。

表 1-13 に、status2 カラムのビット表示をリストします。

表 1-13: sysindexes テーブル status2 カラムのステータス・ビット

10 進数	16 進数	ステータス
1	0x1	インデックスは外部キー制約をサポートする。
2	0x2	インデックスは主キー制約および一意性制約をサポートする。
4	0x4	インデックスに IDENTITY カラムが含まれている。
8	0x8	制約名の指定なし。
16	0x10	テーブル、インデックス、テキスト・チェーンに対して、大容量 I/O (プリフェッチ) が有効でない。
32	0x20	テーブル、インデックス、テキスト・チェーンに対して、MRU (最も最近に使用された) キャッシュ方式が有効でない。
64	0x40	そのテーブルに対して昇順挿入が設定されている。
256	0x0100	インデックスは事前にソートされ、新しいエクステントにコピーする必要はない。
512	0x0200	テーブルは、クラスタード・インデックスを持つデータオンリーロック・テーブル。
8192	0x2000	データオンリーロック・テーブル上のインデックスが疑わしい (suspect 状態)。
32768	0x8000	インデックスは関数ベース。

インデックス

- id、indid カラムに、ユニーク・クラスタード・インデックスが作成されています。

sysinstances

説明

インスタンスのステータスをレポートする偽のテーブルです。`sysinstances` には、クラスタ設定に定義されているインスタンスごとに 1 つのローが含まれます。`sysinstances` には、Cluster Edition に固有の情報が格納されます。

`sysinstances` は偽のテーブルですが、`set system_view` の設定の影響を受けず、`system_view` の設定にかかわらずインスタンスごとに常に 1 つのローを返します。

カラム

`sysinstances` のカラムは次のとおりです。

カラム名	データ型	説明
id	tiny int	インスタンスの ID。
name	varchar(30)	インスタンスの名前。
state	char(17)	インスタンスのステータス (online、offline、joining、leaving、および initiating のいずれか)。
hostname	varchar(255)	このインスタンスを実行しているオペレーティング・システム・ホストの名前。
starttime	datetime	インスタンスが開始された日付と時刻。
connections_active	int	インスタンス上のアクティブな接続の数。
engines_online	smallint	このインスタンスのオンライン・エンジンの数。

インデックス

なし

sysjars

すべてのデータベース

説明

sysjars には、データベースに必要な Java アーカイブ (JAR) ファイルごとに 1 つのローが含まれています。

JAR ファイル、Java クラス、Java データ型の詳細については、『Adaptive Server Enterprise における Java』を参照してください。

カラム

sysjars のカラムは次のとおりです。

名前	データ型	説明
jid	int	JAR の ID。
jstatus	int	内部ステータス情報。未使用。
jname	varchar(255) null	JAR の名前。
jbinary	image null	JAR の内容。Java クラス。

インデックス

- jid カラムに、ユニーク・クラスタード・インデックスが作成されています。
- jname カラムに、ユニーク・ノンクラスタード・インデックスが作成されています。

syskeys

すべてのデータベース

説明 syskeys には、プライマリ・キー、外部キー、または共通キーごとに 1 つのローが含まれています。

カラム syskeys のカラムは次のとおりです。

名前	データ型	説明
id	int	オブジェクト ID。
type	smallint	レコード・タイプ。有効な値は次のとおり。 <ul style="list-style-type: none">• 1 = プライマリ・キー• 2 = 外部キー• 3 = 共通キー
depid	int null	従属オブジェクト ID。
keycnt	int null	null 以外のキーの数。
size	int null	予約済み。
key1 ~ key8	smallint null	カラム ID。
depkey1 ~ depkey8	smallint null	カラム ID。
spare1	smallint	予約済み。

インデックス

- id カラムに、クラスタード・インデックスが作成されています。

syslanguages

master データベースのみ

説明 syslanguages テーブルには、Adaptive Server で認識される言語ごとに 1 つのローが含まれます。us_english 言語は syslanguages には格納されませんが、常に Adaptive Server で利用できます。

カラム syslanguages のカラムは次のとおりです。

名前	データ型	説明
langid	smallint	ユニークな言語 ID。
dateformat	char(3)	“dmy” など、日付の順序。
datefirst	tinyint	週の最初の曜日。月曜日は 1、火曜日は 2、以降同様に続き、日曜日は 7。
upgrade	int	この言語の最後のアップグレードの Adaptive Server バージョン。
name	varchar(30)	“french” など、公式言語名。
alias	varchar(30) null	“français” など、代替言語名。
months	varchar(251)	カンマで区切られた、省略しない月名のリスト。1 月から 12 月の順番で、各月名は最大 20 文字。
shortmonths	varchar(119)	カンマで区切られた、省略した月名のリスト。1 月から 12 月の順番で、各月名は最大 9 文字。
days	varchar(216)	カンマで区切られた曜日名のリスト。月曜から日曜の順番で、各曜日名は最大 30 文字。

インデックス

- langid カラムに、ユニーク・クラスタード・インデックスが作成されています。
- name カラムに、ユニーク・ノンクラスタード・インデックスが作成されています。
- alias カラムに、ユニーク・ノンクラスタード・インデックスが作成されています。

syslisteners

master データベースのみ

説明 syslisteners には、現在の Adaptive Server と接続できるネットワーク・プロトコルごとに 1 つのローが含まれています。ユーザ・アプリケーションまたはクライアント・アプリケーションがテーブルにクエリを実行すると、Adaptive Server は syslisteners を動的に構築します。

カラム syslisteners のカラムは次のとおりです。

名前	データ型	説明
net_type	char(32)	ネットワーク・プロトコル。
address_info	char(255)	ネットワーク上の Adaptive Server をユニークに識別する情報。通常は現在の Adaptive Server 名と、プロトコルに対するサーバのポート番号などの識別番号。
spare	tinyint	未使用。
nodeid	tinyint null	今後のために予約済み (クラスタ環境では使用不可)。
instanceid	tinyint	インスタンスの ID (クラスタ環境のみで使用可能)。

syslocks

master データベースのみ

説明 `syslocks` には、アクティブなロックについての情報が含まれており、ユーザがクエリを実行するときに動的に構築されます。`syslocks` の更新はできません。

カラム `syslocks` のカラムは次のとおりです。

名前	データ型	説明
id	int	テーブル ID。
dbid	smallint	データベース ID。
page	int	ページ番号。
type	smallint	ロックのタイプ (<code>type</code> カラムのビット値については、表 1-14 を参照)。
spid	smallint	ロックを保持するプロセス ID。
	int (Cluster Edition)	
class	varchar(30)	存在する場合は、このロックに関連するカーソルの名前。
fid	smallint (Cluster Edition の場合は int)	ロックが属するファミリ (コーディネーティング・プロセスとそのワーカー・プロセス)。fid の値については、表 1-15 を参照。
context	tinyint	ロック要求のコンテキスト・タイプ。context の値については、表 1-16 を参照。
row	smallint	ロー番号。
loid	int	ユニーク・ロック所有者 ID。
partitionid	int null	パーティション ID。
nodeid	tinyint null	今後のために予約済み (クラスタ環境では使用不可)。
instanceid	tinyint	インスタンスの ID (クラスタ環境のみで使用可能)。

注意 Cluster Edition ではデータ型にこの変更が加えられたため、監査テーブルをアップグレードする前にアーカイブおよびトランケートすることを強くおすすめします。それによって、`sybsecurity` データベース内の領域不足によるアップグレードの失敗を低減できます。

表 1-14 に、`type` カラムのビット表示をリストします。

表 1-14: syslocks テーブルにある type 制御ビット

10 進数	16 進数	ステータス
1	0x1	排他テーブル・ロック。
2	0x2	共有テーブル・ロック。
3	0x3	意図的排他ロック。
4	0x4	意図的共有ロック。
5	0x5	排他ページ・ロック。
6	0x6	共有ページ・ロック
7	0x7	更新ページ・ロック。
8	0x8	排他ロー・ロック。
9	0x9	共有ロー・ロック。
10	0xA	更新ロー・ロック。
11	0xB	共有ネクスト・キー・ロック。
256	0x100	ロックが別のプロセスをブロックしている。
512	0x200	デマンド・ロック。

表 1-15 に、fid カラムの値をリストします。

表 1-15: syslocks テーブルの fid カラムの値

値	解釈
0	spid で示されるタスクは、逐次で文を実行しているシングル・タスクである。
0 以外	ロックを保持しているタスク (spid) は、並列で文を実行しているファミリのメンバである。値が spid に等しい場合、そのタスクは並列でクエリを実行しているファミリのコーディネーティング・プロセスであることを示す。

表 1-16 に、context カラムの値をリストします。

表 1-16: syslocks テーブルの context カラムの値

値	解釈
NULL	このロックを持つタスクは、逐次でクエリを実行しているタスクか、トランザクションの独立性レベル 1 によって並列で実行されているクエリである。
0x1	ロックを持っているタスクは、クエリが終了するまでロックを保持する。ロックのコンテキスト値は、次の場合に FAM_DUR (0x1H) になる可能性がある。 <ul style="list-style-type: none"> • ロックが並列クエリの一部として保持されているテーブルロックである。 • ロックがトランザクション独立性レベル 3 でワーカー・プロセスによって保持されている。 • ロックが並列クエリのワーカー・プロセスによって保持され、トランザクションの間は保持される必要がある。
0x2	直列化可能な読み込みタスクによって保持された範囲ロック。
0x4	無限キー・ロック。
0x8	全ページ・ロック・テーブルのインデックス・ページ上で取得されたロック。
0x10	ローを削除するために取得したページまたはロー上のロック。
0x20	シュリンクまたはスプリット操作時にインデックス・ページ上で取得されたアドレス・ロック。
0x40	繰り返し読み出しを実行しているトランザクションによって保持されている意図的ロック。データオンリー・ロック・テーブル上の意図的共有ロックおよび意図的排他ロックの場合に有効。

sysloginroles

master データベースのみ

説明

`sysloginroles` には、システム役割を持つサーバ・ログインのインスタンスごとに 1 つのローが含まれています。それぞれのログインに所有されている役割ごとに、1 つのローが追加されます。たとえば、1 人のサーバ・ユーザに 3 つの役割 (`sa_role`、`sso_role`、および `oper_role`) が付与されている場合、そのユーザのシステム・ユーザ ID (`suid`) に関連付けられている `sysloginroles` に 3 つのローが追加されます。

カラム

`sysloginroles` のカラムは次のとおりです。

名前	データ型	説明
<code>suid</code>	<code>int</code>	サーバ・ユーザ ID。
<code>srid</code>	<code>int</code>	サーバの役割 ID。次のうちのいずれか。 <ul style="list-style-type: none"> • 0 = <code>sa_role</code> • 1 = <code>sso_role</code> • 2 = <code>oper_role</code> • 4 = <code>navigator_role</code> • 5 = <code>replication_role</code> • 6 = 現在未使用 • 7 = <code>dtm_tm_role</code> • 8 = <code>ha_role</code> • 8 = <code>ha_role</code> • 9 = 内部使用 • 10 = <code>mon_role</code> • 11 = <code>js_admin_role</code> • 12 = <code>messaging_role</code> • 13 = <code>js_client_role</code> • 14 = <code>js_user_role</code> • 15 = <code>web_services</code>
<code>status</code>	<code>smallint</code>	さまざまなサーバの役割がログイン時にデフォルトに設定されるかどうかを示すステータス・ビット。 <ul style="list-style-type: none"> • 0 = デフォルトでオフ • 1 = デフォルトでオン

ステータス・ビットの変更 `sp_modifylogin` を使用してステータス・ビットを変更した場合、変更内容を有効にするには、一度ログアウトしてから再度ログインする必要があります。結果をすぐに確認するには、`set role role_name off` を使用します。

インデックス

- `suid` カラムに、クラスタード・インデックスが作成されています。

syslogins

master データベースのみ

説明 syslogins には、有効な Adaptive Server ユーザ・アカウントごとに 1 つのローが含まれています。

カラム syslogins のカラムは次のとおりです。

名前	データ型	説明
suid	int	サーバ・ユーザ ID。
status	smallint	アカウントのステータス (表 1-17 を参照)。
accdate	datetime	totcpu および totio が最後にクリアされた日付。
totcpu	int	ログインによって累積された CPU 時間。
totio	int	ログインによって累積された I/O。
spacelimit	int	予約済み。
timelimit	int	予約済み。
resultlimit	int	予約済み。
dbname	sysname null	接続が確立されたときユーザが使用するデータベースの名前。
name	sysname	ユーザのログイン名。
password	varbinary(128) null	ユーザ・パスワードの 1 方向ハッシュ。 注意 syslogins.password の内容は、sp_passwordpolicy allow password downgrade の値によって異なります。
language	varchar(30) null	ユーザのデフォルト言語。
pwdate	datetime null	最後にパスワードを変更した日付。
audflags	int null	ユーザの監査設定。
fullname	varchar(30) null	ユーザのフルネーム。
srvname	varchar(30) null	AUTOCONNECT フラグが on の場合は、パススルー接続を確立しなければならないサーバの名前。
logincount	smallint null	ログインを失敗した回数。成功した場合に 0 にリセット。
procid	int null	sp_modifylogin 内の login script オプションで登録されているログイン・トリガを格納する。
lastlogindate	datetime	ユーザの前回ログイン時のタイムスタンプ。
crdate	datetime	ログインが作成されたときのタイムスタンプ。
locksuid	int	ログインのロックを行ったサーバ・ユーザ ID (suid)。

名前	データ型	説明
lockreason	int	<p>ロックの理由。次のいずれか。</p> <ul style="list-style-type: none"> • NULL – アカウントはロックされていない。 • 0 – sp_locklogin の実行により locksuid がアカウントをロックした。 • 1 – sp_locklogin 'all', 'lock', 'ndays' の実行により locksuid が非アクティブなアカウントをロックした。 • 2 – 失敗したログイン試行回数が max failed logins 値に達したために Adaptive Server がアカウントをロックした。 • 3 – パスワード・ダウングレード期間が終了し、ログインまたは役割が SHA-256 に移行しなかったため、locksuid がアカウントをロックした。
lockdate	datetime	ログインがロックされたときのタイムスタンプ。

Adaptive Server の配布メディアにある `syslogins` には、ユーザ名が “sa”、suid が 1、パスワードが null のエントリが含まれています。また、パスワードが発行されていない “probe” というエントリも含まれています。ログインの “probe” とユーザの “probe” は、2 フェーズ・コミットのプローブ・プロセス用です。このプロセスは、Adaptive Server へのアクセスにチャレンジ/応答メカニズムを使用します。

表 1-17 に、status カラムのビット表示をリストします。

表 1-17: `syslogins` テーブルのステータス制御ビット

10 進数	16 進数	ステータス
2	0x2	アカウントはロックされている。
4	0x4	パスワードの有効期限が切れている。
8	0x8	ユーザが RepSrv 権限を持っている。
16	0x10	OMNI: 自動接続モードが有効になっている。
32	0x20	Adaptive Server 内部認証メカニズム (<code>syslogins</code>) を使用できる。
64	0x40	LDAP 外部認証を使用できる。
128	0x80	PAM 外部認証を使用できる。
256	0x100	Kerberos 外部認証を使用できる。

インデックス

- `suid` カラムに、ユニーク・クラスタード・インデックスが作成されています。
- `name` カラムに、ユニーク・ノンクラスタード・インデックスが作成されています。

syslogs

すべてのデータベース

説明

syslogs には、トランザクション・ログが格納されています。リカバリとロールフォワードを行うために Adaptive Server によって使用されます。ユーザが使用するためのテーブルではありません。

syslogs の内容を削除、追加、更新することはできません。データを変更するオペレーションはすべてログが取られるため、**syslogs** を変更する前に、変更内容のログが取られます。つまり、**syslogs** を変更するオペレーションが **syslogs** にローを追加し、このオペレーションのログをとるためのローが **syslogs** に新たに追加されることになり、同じ作業が繰り返されて無限ループが発生します。ループはデータベースがいっぱいになるまで続きます。

カラム

syslogs のカラムは次のとおりです。

名前	データ型	説明
xactid	binary(6)	トランザクション ID。
op	tinyint	更新オペレーション番号。

syslogshold

master データベースのみ

説明 syslogshold には、各データベースの最も古いアクティブ・トランザクション (存在する場合) と、トランザクション・ログに対する Replication Server のトランケーション・ポイント (存在する場合) についての情報が含まれています。ただし、これは通常のテーブルではありません。ユーザがクエリを実行するときに動的に構築されます。syslogshold は更新できません。

カラム syslogshold のカラムは次のとおりです。

名前	データ型	説明
dbid	smallint	データベース ID。
reserved	int	未使用。
spid	smallint	最も古いアクティブ・トランザクションを所有するユーザのサーバ・プロセス ID (Replication Server では常に 0)。
クラスタ環境では int		
page	int	最も古いトランザクションで定義された syslogs のアクティブ部分の開始ページ番号 (Replication Server の場合は syslogs 内のトランケーション・ページ)。
xactid	binary(6)	最も古いアクティブ・トランザクションの ID (Replication Server では常に 0x000000)。
masterxactid	binary(6)	複数データベースのトランザクションに対するトランザクションのマスター・トランザクション (存在する場合) の ID。それ以外は 0x000000 (Replication Server では常に 0x000000)。
starttime	datetime	トランザクションを開始した日付と時間 (Replication Server の場合はトランケーション・ポイントが設定された日付と時間)。
name	char(67)	最も古いアクティブ・トランザクションの名前。これは、 begin transaction で定義される名前。ただし、 begin transaction で値が指定されない場合は “\$user_transaction”、ANSI 連鎖モードで開始された暗黙のトランザクションでは “\$chained_transaction”。Adaptive Server によって開始された内部トランザクションは、ドル記号 (\$) で始まり、オペレーションに由来した名を持つ (Replication Server では “\$replication_truncation_point”)。
xloid	int null	ロック所有権 ID。所有者がタスクの場合は spid に基づく ID、所有者がトランザクションの場合は xdes に基づく ID。

注意 Cluster Edition ではデータ型にこの変更が加えられたため、監査テーブルをアップグレードする前にアーカイブおよびトランケートすることを強くおすすめします。それによって、**sybsecurity** データベース内の領域不足によるアップグレードの失敗を低減できます。

sysmessages

master データベースのみ

説明 sysmessages には、Adaptive Server が返すシステム・エラーまたは警告ごとに 1 つのローが含まれています。Adaptive Server は、画面にエラーの内容を表示します。

カラム sysmessages のカラムは次のとおりです。

名前	データ型	説明
error	int	ユニークなエラー番号。
severity	smallint	エラーの重大度。
dlevel	smallint	予約済み。
description	varchar(1024)	パラメータ用のプレースホルダを持つエラーの説明。
langid	smallint null	言語。us_english の場合 null。
sqlstate	varchar(5) null	エラーの SQLSTATE 値。

インデックス

- error、dlevel カラムに、クラスタード・インデックスが作成されています。
- error、dlevel、langid カラムに、ノンクラスタード・インデックスが作成されています。

sysmonitors

master データベースのみ

説明

sysmonitors には、モニタ・カウンタごとに 1 つのローが含まれています。

カラム

sysmonitors のカラムは次のとおりです。

名前	データ型	説明
field_name	char(79)	カウンタの名前。
group_name	char(25)	このカウンタが属するグループ。
field_id	smallint	ローのユニークな識別子。
value	int	カウンタの現在の値。
description	varchar(255) null	カウンタの説明。未使用。
nodeid	tinyint null	今後のために予約済み (クラスタ環境では使用不可)。
instanceid	tinyint	インスタンスの ID (クラスタ環境のみで使用可能)。

sysobjects

すべてのデータベース

説明

sysobjects には、テーブル、ビュー、ストアド・プロシージャ、拡張ストアド・プロシージャ、ログ、ルール、デフォルト、トリガ、検査制約、参照制約、計算カラム、関数ベース・インデックス・キー、テンポラリ・オブジェクト (**tempdb** のみ)、その他の形式のコンパイル済みオブジェクトごとに 1 つのローが含まれています。さらに、オブジェクトの **type** が N の場合は、パーティションの条件 ID ごとに 1 つのローが含まれます。

sysobjects には、タイプが EK (暗号化キー) の各キーのエントリが含まれます。

データベース間でのキー参照に関しては、**syscolumns.enchrdate** と **sysobjects.crdate** が一致します。

sysencryptkeys の **enckeyid** は、**sysobjects** の **id** カラムと一致します。

カラム

sysobjects のカラムは次のとおりです。

名前	データ型	説明
name	varchar(255) not null	オブジェクト名。
id	int	オブジェクト ID。
uid	int	オブジェクト所有者のユーザ ID。
type	char(2)	次に示すオブジェクト・タイプのいずれか。 <ul style="list-style-type: none"> • C – 計算カラム • D – デフォルト • DD – 復号化デフォルト • F – SQLJ 関数 • L – ログ • N – パーティション条件 • P – Transact-SQL プロシージャまたは SQLJ プロシージャ • PR – 準備オブジェクト (動的 SQL により作成) • R – ルール • RI – 参照制約 • S – システム・テーブル • TR – トリガ • U – ユーザ・テーブル • V – ビュー • XP – 拡張ストアド・プロシージャ
userstat	smallint	アプリケーション従属型の情報 (10 進 32768 [= 16 進 0x8000]) はプロシージャがレポートであることを Data Workbench® に通知する)。
sysstat	smallint	内部ステータス情報 (10 進 256 [= 16 進 0x100]) は読み込み専用テーブルであることを示す)。
indexdel	smallint	オブジェクトのスキーマの変更を再度数え、 schemactnt を更新する。

名前	データ型	説明
schemactnt	smallint	オブジェクトのスキーマの変更数 (ルールやデフォルトが追加されると 1 ずつ増える)。
sysstat2	int	追加の内部ステータス情報 (表 1-19 を参照)。
sysstat3	unsigned smallint	追加の内部ステータス情報 (表 1-20 を参照)。
crdate	datetime	オブジェクトを作成した日付。
expdate	datetime	予約済み。
deltrig	int	エントリがテーブルの場合は、削除トリガのストアド・プロシージャ ID を示す。エントリがトリガの場合は、テーブル ID を示す。
instrig	int	エントリがテーブルの場合は、テーブルの挿入トリガのストアド・プロシージャ ID を示す。
updtrig	int	エントリがテーブルの場合は、テーブルの更新トリガのストアド・プロシージャ ID を示す。
seltrig	int	予約済み。
ckfirst	int	テーブルの最初の検査制約の ID。
cache	smallint	予約済み。
audflags	int null	オブジェクトの監査設定。
objspare	smallint	予備。
versions	binary(6) null	このオブジェクトに対して最後に行われたスキーマ変更のバージョン・タイムスタンプ (Replication Server で使用される)。
loginame	varchar(30) null	オブジェクトを作成したユーザのログイン名。
identburnmax	numeric(17) null	このオブジェクトにおいて identity カラムで消去された最大値。
<p>注意 identburnmax カラムは内部フォーマットで保管されます。値が必要な場合は、identity_burn_max() 関数を使用してください。</p>		
spacestate	smallint null	内部でのみ使用。
erlchgts	binary(8) null	内部でのみ使用。

表 1-18 に、sysstat カラムのビット表示をリストします。

表 1-18: sysobjects テーブルの sysstat 制御ビット

10 進数	16 進数	オブジェクト・タイプ	説明
0	0x0	O_ANY	すべての無効なオブジェクト。
1	0x1	O_SYSTEM	システム・オブジェクト。
2	0x2	O_VIEW	ビュー。
3	0x3	O_USER	ユーザ・オブジェクト。
4	0x4	O_PROC	ストアド・プロシージャ。
5	0x5	O_LOG	ログ。
6	0x6	O_DEFAULT	デフォルト値の仕様。
7	0x7	O_DOMAIN	ドメイン・ルール。
8	0x8	O_TRIGGER	トリガ・プロシージャ。
9	0x9	O_REFERENCE	参照整合性制約。

10 進数	16 進数	オブジェクト・タイプ	説明
10	A	O_CHECK	検査制約。
11	B	O_XTYPE	拡張タイプ。
12	C	O_FUNC	ストアド・ファンクション。
		O_TYPE_MAX	O_FUNC 新しいタイプの追加時に、オブジェクト・タイプの最大値を更新する。print ルーチンで O_TYPE_MAX を使用すると、このフィールドの #define 値の文字列を出力できる。 詳細については、print ルーチンの useful/statbits.c とマクロ PRYPESTR の使用方法を参照。
16	0x10	O_CLUST	クラスタード・インデックスがある。
32	0x20	O_NONCLUST	ノンクラスタード・インデックスがある。
		OBJ_FOR_SYSDEPENDS(obj_type)	次のオブジェクト・タイプの作成または削除時に、オブジェクトが sysdepends 内にエントリを必要とするかどうかを検査する。 <ul style="list-style-type: none"> • O_PROC • O_TRIGGER • O_VIEW • O_DEFAULT • O_DOMAIN • O_FUNC
64	0x40	O_LOGGED	オブジェクトのログを取る。 次のビットは、オーバーロードされ、テーブルとストアド・プロシージャで異なる意味を持つ。この情報は、print ルーチン prOBJSTAT_OBJSYSSTAT() で復号化される。
64	0x40	O_PROC_SUBSCRIBABLE	ストアド・プロシージャは、サブスクライブ可能。 Replication Server は、O_LOGGED ビットのオーバーロードをサポートする。オブジェクトがストアド・プロシージャの場合、O_PROC_SUBSCRIBABLE はストアド・プロシージャがサブスクライブ可能かどうかを示すために使用される。
128	0x80	O_IN_CREATE	オブジェクトを作成中。
256	0x100	O_READONLY	オブジェクトに疑わしいインデックスがあるため、dbcc reindex を実行するまで読み取り専用としてのみ使用可能。
512	0x200	O_SUSPECT	リカバリによって損傷の可能性があるというフラグが付けられたオブジェクト。dbcc を実行する必要がある。opentable による検査済み。
1024	0x400	O_FAKE	オブジェクトは「偽」。つまり、オブジェクトは tempdb 内にあり、これを使用するすべてのクエリ・ステップで再定義される。

10 進数	16 進数	オブジェクト・タイプ	説明
2048	0x800	O_EXTTABLE	オブジェクトは、Stratus VOS ファイルなどの外部テーブル。
4096	0x1000	O_RAMBOIX	インデックスを再作成するシステム・テーブルをタグ付ける。
8192	0x2000	O_TEXTIMAGE	text または image フィールドを含むオブジェクト。
16384	0x4000	O_TABNOLOG	未使用。
32768	0x8000	O_REPLICATED	テーブルまたはプロシージャは複写される。

表 1-19 に、sysstat2 カラムのビット表示をリストします。

表 1-19: sysobjects テーブルの sysstat2 制御ビット

10 進数	16 進数	ステータス
1	0x1	テーブルに参照制約がある。
2	0x2	テーブルに外部キー制約がある。
4	0x4	テーブルに1つ以上の検査制約がある。
8	0x8	テーブルに主キー制約がある。
16	0x10	ストアド・プロシージャは連鎖トランザクション・モードのときだけ実行可能である。
32	0x20	ストアド・プロシージャはすべてのトランザクション・モードで実行可能である。
64	0x40	テーブルに IDENTITY フィールドがある。
128	0x80	未使用。
256	0x100	未使用。
512	0x200	テーブルに変長カラムがない。
1024	0x400	テーブルがリモートである。
2048	0x800	テーブルが existing キーワードを使用して作成されたプロキシ・テーブルである。
4096	0x1000	未使用。
8192	0x2000	テーブルが全ページ・ロック・スキームを使用する。
16384	0x4000	テーブルがデータページ・ロック・スキームを使用する。
32768	0x8000	テーブルがデータロー・ロック・スキームを使用する。
65536	0x10000	テーブルは、バージョン 11.9 以降のサーバで作成された。
131072	0x20000	テーブルにクラスタード・インデックスがある。
262144	0x40000	オブジェクトは、Embedded SQL プロシージャを表す。
524288	0x80000	未使用。
16777216	0x1000000	オブジェクトは、アクセス・ルールを表す。
33554432	0x2000000	オブジェクトは、SQLJ ストアド・プロシージャを表す。
67108864	0x4000000	オブジェクトは、OR アクセス・ルールを表す。
1073741824	0x40000000	テーブルに1つまたは複数の関数ベース・インデックスがある。
2147483648	0x80000000	オブジェクトに拡張インデックスがある。

表 1-20 に、sysstat3 カラムのビット表示をリストします。

表 1-20: sysobjects テーブルの sysstat3 制御ビット

10 進数	16 進数	ステータス
32768	0x8000	テーブルは増分転送に関与する。

インデックス

- id カラムに、ユニーク・クラスタード・インデックスが作成されています。
- name, uid カラムに、ノンクラスタード・インデックスが作成されています。

sysoptions

すべてのデータベース

説明 sysoptions は、sp_options によるクエリの対象となる新しい偽のテーブルです。sysoptions へのクエリを実行する場合、ローの名前には大文字と小文字の区別があります。

カラム

名前	データ型	属性	説明
spid	int		プロセス ID を格納します。
name	varchar(100)		オプションの名前を格納します。
category	varchar(100)		オプションが属するカテゴリの名前を格納します。
currentsetting	varchar(100)	NULL	オプションの現在の設定を格納します。
defaultsetting	varchar(100)	NULL	オプションのデフォルト設定を格納します。
scope	int		オプションに関する情報を取得するために使用されたビットマップを格納します。ビットの順序は次のとおりです。 <ul style="list-style-type: none"> • Bit 1 – コンパイル済みの time オプション • Bit 2 – ストアド・プロシージャ固有のオプション • Bit 3 – binary オプション

sp_options を使用して sysoptions に対しクエリを実行します。現在値とデフォルト値のデータ型は varchar なので、varchar 値での設定を直接使用できます。integer 値での設定は、型変換後に使用できます。

sysoptions に対するクエリを実行するには、特殊な権限は必要ありません。次に例を示します。

```
select * from sysoptions
where spid = 13
go
```

文字列操作または型変換も使用できます。たとえば、オプションが数値の場合、次を入力すると sysoptions に対しクエリを実行できます。

```
if (isnumeric(currentsetting))
    select@int_val = convert(int, currentsetting)
...
else
    select@char_val = currentsetting
...
```

syspartitionkeys

すべてのデータベース

説明 syspartitionkeys には、テーブルのハッシュ・パーティション、範囲パーティション、リスト・パーティションのパーティション・キーごとに 1 つのローが含まれています。すべてのカラムが null ではありません。

カラム syspartitionkeys のカラムは次のとおりです。

名前	データ型	説明
indid	smallint	インデックスのタイプ。値は次のとおり。 <ul style="list-style-type: none">• 0 = テーブル• 1 = クラスタード・インデックス• >1 = ノンクラスタード・インデックス
id	int	分割されたテーブルのオブジェクト ID。
colid	smallint	分割されたテーブルのパーティション・キーのカラム ID。
position	smallint	キー位置におけるキーの場所。

インデックス

- id, indid, colid カラムに、ユニーク・クラスタード・インデックスが作成されています。

syspartitions

すべてのデータベース

説明

`syspartitions` には、データ・パーティションごとに 1 つのロー、およびインデックス・パーティションごとに 1 つのローが含まれています。

各データベースについて、`syspartitions` には、以下のそれぞれごとに 1 つのローが含まれています。

- 各テーブル・パーティション。indid は 0。
- 各クラスタード・インデックス・パーティション。indid は 1。
- 各ノンクラスタード・インデックス・パーティション。indid は >1。
- 単一パーティション (未分割) の各テーブル。
- 単一パーティション (未分割) の各クラスタード・インデックスまたはノンクラスタード・インデックス。

インデックスがローカルの場合、`partitionid` (データ・パーティション・ロー) と `data_partitionid` (対応するインデックス・ロー) の値は同じです。インデックスがローカルでない場合、`data_partitionid` (インデックス・ロー) の値はゼロ (0) で、`partitionid` (データ・パーティション・ロー) とは等しくありません。

注意 Adaptive Server バージョン 15.0 より前の `syspartitions` テーブルは、名前が `syslices` に変更され、旧式になりました。Adaptive Server バージョン 15.0 では、`syspartitions` が完全に再定義され、データ・パーティションおよびインデックス・パーティションがサポートされました。

カラム

`syspartitions` のカラムは次のとおりです。

名前	データ型	説明
<code>name</code>	<code>varchar(255)</code>	パーティション名。
<code>indid</code>	<code>smallint</code>	全ページロック・テーブルの場合 インデックス ID。 <ul style="list-style-type: none"> • 0 = データ・ページ (テーブル) • 1 = 全ページロック・テーブルのクラスタード・インデックス • >1 および <255 = ノンクラスタード・インデックスまたはデータオンリーロック・テーブルのクラスタード・インデックス • 255 = テキスト・チェーン
<code>id</code>	<code>int</code>	テーブル ID。
<code>partitionid</code>	<code>int</code>	データまたはインデックス・パーティションの ID。
<code>segment</code>	<code>smallint</code>	パーティションがあるセグメントの ID。
<code>status</code>	<code>int</code>	内部ステータス情報。
<code>datoampage</code>	<code>int</code>	データ・パーティションのオブジェクト・アロケーションのページ番号。

名前	データ型	説明
indoampage	int	インデックス・パーティションのオブジェクト・アロケーション・マップのページ番号。
firstpage	int	最初のデータ・ページまたはリーフ・ページのページ番号。
rootpage	int	以下のページ番号。 <ul style="list-style-type: none"> • ルート・ページ (エントリがインデックス・パーティションの場合)。 • 最後のページ (エントリがデータ・パーティションの場合)。
data_partitionid	int	このインデックスがまたがるデータ・パーティションの ID。値は次のとおり。 <ul style="list-style-type: none"> • 0 = テーブル全体にまたがるグローバル・インデックス。 • 1 = ローカル・インデックスのパーティションがまたがるデータ・パーティションのパーティション ID。
crdate	datetime	パーティションの作成日。
cdataptnname	varchar(255) null	データ・パーティションの名前。

- インデックス
- **id**、**indid**、**partitionid** カラムに、ユニーク・クラスタード・インデックスが作成されています。
 - **id**、**indid**、**name** カラムに、ユニーク・ノンクラスタード・インデックスが作成されています。
 - **partitionid**、**indid** カラムに、ユニーク・ノンクラスタード・インデックスが作成されています。

sysprocedures

すべてのデータベース

説明

sysprocedures には、ビュー、デフォルト、ルール、トリガ、プロシージャ、宣言デフォルト、パーティション条件、検査制約、計算カラム、関数ベース・インデックス・キー、その他の形式のコンパイル済みオブジェクトごとにエントリが含まれています。計算カラムまたは関数ベース・インデックス定義を含め、各オブジェクトのシーケンス・ツリーは、バイナリ形式で格納されています。シーケンス・ツリーが 1 つのエントリに収まらない場合は、複数のローに分割されます。sequence カラムはサブローを識別します。

カラム

sysprocedures のカラムは次のとおりです。

名前	データ型	説明
type	smallint	オブジェクトのタイプ (表 1-21 を参照)。
qp_setting	varbinary(6) null	今後のために予約済み。
id	int	オブジェクト ID。
sequence	int	オブジェクトを説明するために複数のローが使用された場合のシーケンス番号。
status	smallint	内部システム・ステータス。
number	smallint	プロシージャがグループ化されている場合のサブ・プロシージャ番号 (プロシージャ・エントリでない場合は 0)。
version	int null	当該のオブジェクトについて、このカタログに格納されているシーケンス・ツリーを作成した Adaptive Server のバージョン。

表 1-21 に、type カラムのビット表示をリストします。

表 1-21: sysprocedures テーブルにある type 制御ビット

10 進数	16 進数	ステータス
1	0x1	エントリはプランを示す (予約済み)。
2	0x2	エントリはツリーを示す。

インデックス

- id、number、type、sequence カラムに、ユニーク・クラスタード・インデックスが作成されています。

sysprocesses

master データベースのみ

説明 `sysprocesses` には Adaptive Server のプロセスについての情報が含まれていますが、このテーブルは、通常のテーブルではありません。ユーザがクエリを実行するときに動的に構築されます。`sysprocesses` の更新はできません。プロセスを強制終了するには `kill` 文を使用します。

カラム `sysprocesses` のカラムは次のとおりです。

名前	データ型	説明
spid	smallint int (Cluster Edition)	プロセス ID。
kpid	int	カーネル・プロセス ID。
enginenum	int	プロセス実行中のエンジン番号。
status	char(12)	プロセス ID ステータス (表 1-22 を参照)。
suid	int	コマンドを実行したユーザのサーバ・ユーザ ID。
hostname	varchar(30) null	ホスト・コンピュータ名。
program_name	varchar(30) null	フロントエンド・モジュール名。
hostprocess	varchar(30) null	ホスト・プロセスの ID 番号。
clientport	unsigned smallint	<ul style="list-style-type: none"> クライアント・プロセスのクライアント・ポート番号を表示する。 システム・プロセスの場合は 0 を表示する。
cmd	varchar(30) null	現在実行中のコマンドまたはプロセス。if または while のループなどの条件文が評価されると、cond を返す。
cpu	int	CPU の累積占有時間 (チック単位)。
physical_io	int	現在のコマンドのディスク読み込みと書き込み数。
memusage	int	プロセスに割り当てられたメモリの容量。
blocked	smallint int (Cluster Edition)	ブロックしているプロセスがある場合のプロセス ID。
dbid	smallint	データベース ID。
uid	int	コマンドを実行したユーザの ID。
gid	int	コマンドを実行したユーザのグループ ID。
tran_name	varchar(64) null	アクティブなトランザクションの名前。
time_blocked	int null	ブロックされた時間 (秒単位)。
network_pktsz	int null	現在の接続のネットワーク・パケット・サイズ。
fid	smallint int (Cluster Edition)	ワーカー・プロセスの親プロセスのプロセス ID。
execclass	varchar(30) null	プロセスがバインドされている実行クラス。
priority	varchar(10) null	プロセスに対応する基本優先度。
affinity	varchar(30) null	プロセスの結び付きが設定されたエンジンの名前。
id	int null	現在実行されているプロシージャのオブジェクト ID (プロシージャが実行されていない場合には、0)。

名前	データ型	説明
stmtnum	int null	実行されているプロシージャ内の現在の文番号 (プロシージャが実行されていない場合は、SQL バッチ文の番号)。
linenum	int null	実行されているストアド・プロシージャ内の現在の文の行番号 (プロシージャが実行されていない場合は、現在の SQL バッチ文の行番号)。
origsuid	int null	この値が NULL でない場合は、コマンドを実行したユーザを代行するために <code>set proxy</code> または <code>set session authorization</code> を実行している <code>origsuid</code> の <code>suid</code> を持つユーザ。
block_xloid	int null	トランザクションをブロックしているロックのユニーク・ロック所有者 ID。
clientname	varchar(30) null	オプション - 現在のセッションで識別されているユーザの名前。 注意 Adaptive Server は 1 つ以上のスペースを <code>clientname</code> 、 <code>clienthostname</code> 、および <code>clientappliance</code> カラムに自動的に格納します。したがって、これら 3 つのカラムのいずれかに対して “is null” を使用するクエリでは、予期する結果セットが返されません。
clienthostname	varchar(30) null	オプション - 現在のセッションで認知されているホストの名前。
clientappliance	varchar(30) null	オプション - 現在のセッションで認知されているアプリケーションの名前。
sys_id	smallint null	コンパニオン・ノードのユニーク ID。
ses_id	int null	個々のクライアント・セッションのユニーク ID。
loggedindatetime	datetime null	クライアントが Adaptive Server に接続した日時を示す。詳細については、『システム管理ガイド』の「第 11 章 ユーザ・パーミッションの管理」の「ロー・レベル・アクセス制御」を参照。
ipaddr	varchar(64) null	ログインしたクライアントの IP アドレス。詳細については、『システム管理ガイド』の「第 11 章 ユーザ・パーミッションの管理」の「ロー・レベル・アクセス制御」を参照。
nodeid	tinyint null	今後のために予約済み (クラスタ環境では使用不可)。
instanceid	tinyint	インスタンスの ID (クラスタ環境のみで使用可能)。
pad	smallint	位置合わせのために追加されるカラム (クラスタ環境のみで使用可能)。
lcid	int	クラスタの ID。

注意 Cluster Edition ではデータ型にこの変更が加えられたため、監査テーブルをアップグレードする前にアーカイブおよびトランケートすることを強くおすすめします。それによって、`sybsecurity` データベース内の領域不足によるアップグレードの失敗を低減できます。

表 1-22 に、status カラムの値をリストします。

表 1-22: sysprocesses の status カラム値

ステータス	意味
alarm sleep	アラームによるプロセスのウェイクアップを待機中 (ユーザが waitfor delay コマンドを実行した)。
background	ユーザ・プロセスではなく Adaptive Server によって実行される、スレッショルド・プロシージャなどのプロセス
infected	サーバが深刻なエラー条件を検出した。これはめったに発生しない。
latch sleep	ラッチ取得を待機中。
lock sleep	ロック取得の待機中。
PLC sleep	ユーザ・ログ・キャッシュのアクセスを待機中。
recv sleep	ネットワーク読み込みの待機中。
remote i/o	リモート・サーバで I/O を実行中。
runnable	実行可能なプロセスのキュー内にある。
running	サーバ・エンジンの 1 つで実行中。
send sleep	ネットワーク送信の待機中。
sleeping	ディスク I/O やその他のリソースで待機中 (プロセスが実行されているが、ディスク I/O で多量のタスクが行われていることを示す場合がよくある)。
stopped	プロセスは停止している。
sync sleep	ファミリー内の他のプロセスから、同期メッセージが届くのを待機中。

sysprotects

すべてのデータベース

説明 sysprotects には、ユーザ、グループ、役割に付与または取り消されたパーミッションの情報が含まれています。

カラム sysprotects のカラムは次のとおりです。

名前	データ型	説明	名前
id	int	パーミッションが適用されるオブジェクトの ID。create table または create default を実行する権限を付与されているユーザには id 0 が指定される。	
uid	int	パーミッションが適用されるユーザ、グループ、または役割の ID。	
action	smallint	次のパーミッションのいずれか。 151 = references 167 = set proxy または set session authorization 187 = set statistics on 188 = set statistics off 193 = select 195 = insert 196 = delete 197 = Update 198 = create table 203 = create database 205 = Granted 206 = revoke 207 = create view 221 = create trigger 222 = create procedure 224 = execute 228 = dump database 233 = create default 235 = dump transaction 236 = create rule 253 = connect 282 = delete statistics 317 = dbcc 320 = truncate table 326 = update statistics 347 = set tracing	
protecttype	tinyint	次の値のいずれか。 <ul style="list-style-type: none"> 0 = grant with grant 1 = Granted 2 = revoke 	
columns	varbinary(133)	この select、update、または references パーミッションが適用されるカラムのビット・マップ。ビットは次のいずれかを示す。 <ul style="list-style-type: none"> 0 = すべてのカラムに適用される。 1 = そのカラムに適用される。 NULL = 情報がない。 columns は、set session authorization に対して許可された役割のビットマップでもある。	
grantor	int	付与した人のユーザ ID。付与者がシステム管理者の場合はオブジェクト所有者のユーザ ID。	

インデックス

- id、action、grantor、uid、protecttype カラムに、ユニーク・クラスタード・インデックスが作成されています。

sysquerymetrics

すべてのデータベース

説明

永続的なデータからの個々のクエリに関して集約された履歴クエリ処理の測定基準を示します。モニタリング・テーブルに加え、このカタログからパフォーマンス測定基準情報を使用します。

カラム

sysquerymetrics のカラムは次のとおりです。

名前	データ型	説明
uid	int	ユーザ ID。
gid	int	グループ ID。
hashkey	int	SQL クエリ・テキストにおけるハッシュ・キー。
id	int	ユニークな ID。
sequence	smallint null	SQL のテキストに複数ローが必要な場合の、ローのシーケンス番号。
exec_min	int null	最小実行時間。
exec_max	int null	最大実行時間。
exec_avg	int null	平均実行時間。
elap_min	int null	最小経過時間。
elap_max	int null	最大経過時間。
elap_avg	int null	平均経過時間。
lio_min	int null	最小論理 I/O。
lio_max	int null	最大論理 I/O。
lio_avg	int null	平均論理 I/O。
pio_min	int null	最小物理 I/O。
pio_max	int null	最大物理 I/O。
pio_avg	int null	平均物理 I/O。
cnt	int null	クエリが実行された回数。
abort_cnt	int null	リソース制限を超えたときに Resource Governor によってクエリがアポートされた回数。
qtext	varchar(255) null	クエリ・テキスト。

Adaptive Server リリース 15.0.2 以降では、ユーザ ID 間で共有される測定基準の数が増加しました。その結果、sysquerymetrics (sysqueryplans テーブルのビュー) のエントリ数が減り、異なるユーザ ID 間の同一クエリの測定基準が自動的に統合されるようになりました。

ユーザ名で修飾されない、クエリ内のすべてのテーブル名を DBO (データベース所有者) が所有している場合、sysquerymetrics のユーザ ID (uid) は 0 です。

たとえば、テーブル t1 の所有者は DBO であるが、さまざまなユーザがこのテーブルを共有している場合などが、これに該当します。

```
select * from t1 where c1 = 1
```

Adaptive Server では、**t1** というプライベート・テーブルを持たない、このクエリを実行するすべてのユーザによる **sysquerymetrics** テーブルに対するエントリの **uid** として **0** が使用されます。

次の例では、テーブル **t2** が “user1” によって所有され、修飾されている場合、UID として **0** が使用されます。

```
select * from user1.t2 where c1 = 1
```

ただし、“user1” のみがテーブル **t3** を所有しており、DBO によってこのテーブルが修飾されておらず、所有もされていない場合は、**sysquerymetrics** テーブルに対するエントリで “user1” の UID が使用されます。

```
select * from t3 where c1 = 1
```

sysqueryplans

すべてのデータベース

説明 sysqueryplans には、抽象クエリ・プランごとに複数のローが含まれています。データロー・ロックを使用します。

カラム sysqueryplans のカラムは次のとおりです。

名前	データ型	説明
uid	int	抽象プランを取得したユーザのユーザ ID。
dbid	int null	今後のために予約済み。
update	datetime null	今後のために予約済み。
sprocid	int null	今後のために予約済み。
hashkey2	int null	今後のために予約済み。
key1	int null	今後のために予約済み。
key2	int null	今後のために予約済み。
key3	int null	今後のために予約済み。
gid	int	保存された抽象プランの抽象プラン・グループ ID。
hashkey	int	SQL クエリ・テキストにおけるハッシュ・キー。
id	int	抽象プランの場合はユニークな ID。
type	smallint	10 = テキスト・カラムにクエリ・テキストが含まれる場合、100 = テキスト・カラムに抽象プラン・テキストが含まれる場合。
sequence	smallint	SQL クエリまたは抽象プランに複数ローが必要な場合は、シーケンス番号。
status	int null	予約済み。
text	varchar(255) null	type = 10 の場合は SQL テキスト、type = 100 の場合は抽象クエリ・プラン・テキスト。

- インデックス**
- uid, gid, hashkey, id, type, sequence カラムに、ユニーク・クラスタード・インデックスが作成されています。
 - id, type, sequence カラムに、ノンクラスタード・インデックスが作成されています。

sysreferences

すべてのデータベース

説明 sysreferences には、テーブルまたはカラムに宣言された参照整合性の制約ごとに 1 つのローが含まれています。

カラム sysreferences のカラムは次のとおりです。

名前	データ型	説明
indexid	smallint	参照先カラムにあるユニーク・インデックスの ID。
constrid	int	sysobjects による制約のオブジェクト ID。
tableid	int	参照元テーブルのオブジェクト ID。
reftabid	int	参照先テーブルのオブジェクト ID。
keycnt	smallint	外部キーのカラム数。
status	smallint	オプションとインジケータ。
frgndbid	smallint null	参照先テーブル (外部キーを持つテーブル) を含むデータベースのデータベース ID。
pmrydbid	smallint	参照先テーブル (プライマリ・キーを持つテーブル) を含むデータベースのデータベース ID。
spare2	int	予約済み。
fokey1 ~ fokey16	smallint	参照元カラムの 1 番目から 16 番目のカラム ID。
refkey1 ~ refkey16	smallint	参照先カラムの 1 番目から 16 番目のカラム ID。
frgndbname	varchar(30) null	参照元テーブル (外部キーを含むテーブル) を持つデータベース名。参照元テーブルが現在のデータベースにある場合は null。
pmrydbname	varchar(30) null	参照先テーブル (プライマリ・キーを含むテーブル) を持つデータベース名。参照先テーブルが現在のデータベースにある場合は NULL。

表 1-23 に sysreferences のステータス・ビットを示します。

表 1-23: sysreferences テーブルのステータス・ビット

10 進数	16 進数	ステータス
2	0x2	参照制約に match full オプションがある。

インデックス

- tableid、frgndbname カラムに、クラスタード・インデックスが作成されています。
- constrid、frgndbname カラムに、ノンクラスタード・インデックスが作成されています。
- reftabid、indexid、pmrydbname カラムに、ノンクラスタード・インデックスが作成されています。

sysremotelogins

master データベースのみ

説明 sysremotelogins には、Adaptive Server でリモート・プロシージャ・コールを実行できるリモート・ユーザごとに1つのローが含まれています。

カラム sysremotelogins のカラムは次のとおりです。

名前	データ型	説明
remoteserverid	smallint	リモート・サーバを示す。
remoteusername	varchar(30) null	リモート・サーバのユーザのログイン名。
suid	int	ローカル・サーバのユーザ ID。
status	smallint	オプションのビットマップ。

インデックス

- remoteserverid、remoteusername カラムに、ユニーク・クラスタード・インデックスが作成されています。

sysresourcelimits

master データベースのみ

説明 sysresourcelimits には、Adaptive Server で定義されるリソース制限ごとに 1 つのローが含まれています。リソース制限は、クエリ、クエリ・バッチ、トランザクションなどを実行するために Adaptive Server ログインやアプリケーションによって使用されるサーバ・リソースの最大値を指定します。

カラム sysresourcelimits のカラムは次のとおりです。

名前	データ型	説明
name	varchar(30) null	ログイン名。
appname	varchar(30) null	アプリケーション名。
rangeid	smallint	systemranges の id カラム。
limitid	smallint	spt_limit_types の id カラム。
enforced	tinyint	spt_limit_types の enforced カラムのサブセット。 <ul style="list-style-type: none"> • 1 = 実行前 • 2 = 実行中 • 3 = その両方
action	tinyint	違反したときの動作。 <ul style="list-style-type: none"> • 1 = 警告を出す • 2 = クエリ・バッチをアボートする • 3 = トランザクションをアボートする • 4 = セッションを強制終了する
limitvalue	int	制限の値。
scope	tinyint	ユーザ制限のスコープ(次のいずれか1つまたは複数を示すビットマップ)。 <ul style="list-style-type: none"> • 1 = クエリ • 2 = クエリ・バッチ • 4 = トランザクション
spare	tinyint	予約済み。

インデックス

- name、appname カラムに、クラスタード・インデックスが作成されています。

sysroles

すべてのデータベース

説明 sysroles はサーバの役割 ID をローカルの役割 ID にマップします。

カラム sysroles のカラムは次のとおりです。

名前	データ型	説明
id	int	サーバの役割 ID (srid)。
lrid	int	ローカルの役割 ID。
type	smallint	未使用。
status	int	未使用。

役割にデータベース・パーミッションを付与するとき、役割のエントリが **sysroles** になければ、Adaptive Server によって **sysroles** にエントリが追加され、ローカル役割 ID (lrid) が **sysroles** のサーバワイドな役割 ID (srid) にマップされます。

インデックス

- lrid カラムに、ユニーク・クラスタード・インデックスが作成されています。

syssecmechs

master データベースのみ

説明

syssecmechs には、Adaptive Server で利用できる各セキュリティ・メカニズムでサポートされているセキュリティ・サービスに関する情報が含まれます。syssecmechs は、インストール時には作成されず、ユーザがクエリを実行したときに動的に構築されます。

カラム

syssecmechs のカラムは次のとおりです。

名前	データ型	説明
sec_mech_name	varchar(30)	セキュリティ・メカニズムの名前。たとえば“NT LANMANAGER”など。
available_service	varchar(30)	セキュリティ・メカニズムがサポートしているセキュリティ・サービスの名前。たとえば“unified login”など。

syssegments

すべてのデータベース

説明

syssegments には、セグメント (ディスクの一部分の集まり) ごとに 1 つのローが含まれています。新しく作成されたデータベースのエントリには、システム・テーブル用のセグメント 0 (system)、トランザクション・ログ用のセグメント 2 (logsegment)、その他のオブジェクト用のセグメント 1 (default) があります。

カラム

syssegments のカラムは次のとおりです。

名前	データ型	説明
segment	smallint	セグメント番号。
name	sysname	セグメント名。
status	smallint null	どのセグメントがデフォルト・セグメントかを示す。

syssservers

master データベースのみ

説明 syssservers には、この Adaptive Server でリモート・プロシージャ・コールを実行できるリモートの Adaptive Server、Backup Server™、または Open Server™ ごとに 1 つのローが含まれています。

カラム syssservers のカラムは次のとおりです。

名前	データ型	説明
srvid	smallint	リモート・サーバの ID 番号 (ローカル使用のみ)。
srvstatus	smallint	オプションのビットマップ (表 1-24 を参照)。
srvstatus2	unsigned int	オプションのビットマップ (表 1-25 を参照)。
srvstat2	unsigned int	サーバ・オプションのビットマップ。
srvname	varchar(30)	サーバの名前。
srvnetname	varchar (255)	サーバの interfaces ファイル名。
srvclass	smallint null	sp_addserver のクラス・パラメータで定義されたサーバのカテゴリ (表 1-26 を参照)。
srvsecmech	varchar(30) null	セキュリティ・メカニズム。
srvcost	smallint null	ネットワーク上でサーバにアクセスするための、ネットワーク・コスト (ミリ秒) を提供。プロキシ・テーブルへのアクセス時に、Adaptive Server クエリ・最適化メカニズムがクエリのコストを評価する場合にのみ使用される。デフォルト値は 1,000 ミリ秒。

表 1-24 に、srvstatus カラムのビット表示をリストします。

表 1-24: srvstatus カラムのステータス制御ビット

10 進数	16 進数	ステータス
0	0x0	タイムアウトが有効である。
1	0x1	タイムアウトが無効である。
2	0x2	ネットワーク・パスワード暗号化が有効である。
4	0x4	リモート・サーバは読み込み専用である。
8	0x8	RPC セキュリティ・モデル A を使用する。
16	0x10	RPC セキュリティ・モデル B を使用する。
64	0x40	メッセージの機密性を使用。
128	0x80	メッセージの整合性を使用。
256	0x100	相互認証。

表 1-25: syssservers テーブルの srvstatus2 制御ビット

10 進数	16 進数	ステータス
1	0x01	完全修飾テーブル名をサポート。
2	0x02	今後のために予約済み。

表 1-26 に、`srvclass` カラムのサーバ・カテゴリをリストします。

表 1-26: syservers テーブルでのサーバ・カテゴリ

srvclass	サーバ・カテゴリ
0	ローカル・サーバ
1	sql_server クラス・サーバ
3	direct_connect クラス・サーバ
4	DB2 クラス・サーバ
6	sds クラス・サーバ
7	Adaptive Server Enterprise クラス・サーバ
8	Adaptive Server Anywhere クラス・サーバ
9	ASIQ クラス・サーバ

インデックス

- `srvuid` カラムに、ユニーク・クラスタード・インデックスが作成されています。
- `srvname` カラムに、ノンクラスタード・インデックスが作成されています。

syssessions

master データベースのみ

説明

syssessions は、高可用性システム内で Adaptive Server が Sybase フェールオーバー用に設定されている場合にのみ使用されます。**syssessions** には、フェールオーバー・プロパティを持つ Adaptive Server に接続するクライアントごとに、1つのローが含まれています。フェールオーバー中に **syssessions** 内にエントリを持つクライアントは、セカンダリ・コンパニオンに移動されます。**syssessions** 内にエントリを持たないクライアントは、フェールオーバー中に削除されます。フェールバック中に **syssessions** 内にエントリを持つクライアントは、プライマリ・コンパニオンに移動されます。フェールバック中に **syssessions** 内にエントリを持たないクライアントは、削除されます。

カラム

syssessions のカラムは次のとおりです。

名前	データ型	説明
sys_id	smallint	コンパニオン・ノードのユニーク ID。
ses_id	int	個々のクライアント・セッションのユニーク ID。
state	tinyint	セッションがアクティブか否かを記述する。
spare	tinyint	今後のために予約済み。
status	smallint	今後のために予約済み。
dbid	smallint	今後のために予約済み。
name	varchar(30) null	syslogins で指定されたクライアントのログイン名と同じ。
nodeid	tinyint null	今後のために予約済み (クラスタ環境では使用不可)。
instanceid	tinyint	インスタンスの ID (クラスタ環境のみで使用可能)。
ses_data	image null	今後のために予約済み。

syslices

すべてのデータベース

説明

syslices には、スライスされたテーブルのスライス (ページ・チェーン) ごとに 1 つのローが含まれています。 **syslices** は、 Adaptive Server がプロセスをアップグレードする間だけ使用されます。アップグレードが完了したら、すべてのデータは削除されます。

注意 Adaptive Server の 15.0 より前のバージョンでは、 **syspartitions** は、分割に関連した情報を格納するカタログの名前でした。 Adaptive Server 15.0 では、この名前が **syslices** に変更され、現在では **syspartitions** は、 Adaptive Server 内のすべての分割に関連したデータを追跡するカタログの名前です。

カラム

syslices のカラムは次のとおりです。

名前	データ型	説明
state	smallint	分割のステータスについての内部情報。
id	int	分割されたテーブルのオブジェクト ID。
partitionid	int	分割の ID 番号。
firstpage	int	分割の最初のページのページ番号。
controlpage	int	分割の制御ページのページ番号。
spare	binary(32)	予約済み。

インデックス

- id、partitionid カラムに、ユニーク・クラスタード・インデックスが作成されています。

sysssrvroles

master データベースのみ

説明 sysssrvroles には、システム役割またはユーザ定義役割ごとに 1 つのローが含まれています。

カラム sysssrvroles のカラムは次のとおりです。

名前	データ型	説明
srid	int	サーバ役割 ID。
name	varchar(30)	役割名。
password	varbinary(128) null	役割のパスワード (暗号化されている)。sso_role のあるユーザのみが判読できる。
pwdate	datetime null	最後にパスワードを変更した日付。
status	smallint null	役割ステータスのビットマップ (表 1-27 を参照)。
logincount	smallint null	ログインを失敗した回数。成功した場合に 0 にリセット。

表 1-27 に、status カラムのビット表示をリストします。

表 1-27: sysssrvroles テーブルのステータス制御ビット

10 進数	16 進数	ステータス
2	0x2	役割はロックされている。
4	0x4	役割の期限切れ。

インデックス

- srid カラムに、ユニーク・クラスタード・インデックスが作成されています。

sysstatistics

すべてのデータベース

説明

sysstatistics には、ユーザ・テーブル上のインデックス・カラムごと、およびパーティションごとに、1つ以上のローが含まれています。インデックスが作成されていないカラムに対してローを含むこともあります。

カラム

sysstatistics のカラムは次のとおりです。

名前	データ型	説明
statid	smallint	予約済み。
id	int	テーブルのオブジェクト ID。
sequence	int	この統計情報に複数ローが必要な場合は、シーケンス番号。
moddate	datetime	このローが最後に修正された日付。
formatid	tinyint	このローによって表記される統計情報のタイプ。
usedcount	tinyint	このローで使用される c0 から c79 までのフィールドの数。
colidarray	varbinary(100)	カラム ID の順序リスト。
c0...c79	varbinary(255)	統計データ。
indid	smallint	パーティションのインデックス ID。
ststatus	smallint	この統計ローのステータス・ビット。取り得る値はローの型によって異なる。
partitionid	int	パーティション ID。
spare2	smallint	今後のために予約済み。
spare3	int	今後のために予約済み。

インデックス

- id、indid、partitionid、statid、colidarray、formatid、sequence カラムに、ユニーク・クラスタード・インデックス **csysstatistics** が作成されています。

sysabstats

すべてのデータベース

説明 sysabstats には、クラスタード・インデックスごとに 1 つのロー、ノンクラスタード・インデックスごとに 1 つのロー、クラスタード・インデックスのないテーブルごとに 1 つのロー、パーティションごとに 1 つのローが含まれています。

カラム sysabstats のカラムは次のとおりです。

名前	データ型	説明
indid	smallint	<ul style="list-style-type: none"> 0 = テーブル 1 = 全ページロック・テーブルのクラスタード・インデックス >1 = ノンクラスタード・インデックスまたはデータオンリーロック・テーブルのクラスタード・インデックス <p>sysabstats には、text または image オブジェクト (255) に関する統計は保持されていません。</p>
id	int	インデックスが設定されたテーブルの ID。
activestatid	smallint	予約済み。
indexheight	smallint	インデックスの高さ。indid が 1 より大きい場合に維持される。
leafcnt	int	インデックスのリーフ・ページ数。indid が 1 より大きい場合に維持される。
pagecnt	int	テーブルまたはインデックス内のページ数。
rowcnt	float	テーブル内のローの数。indid が 0 または 1 の場合に維持される。
forwrowcnt	float	転送されたローの数。indid が 0 または 1 の場合に維持される。
delrowcnt	float	削除されたローの数。
dpagecrnt	float	テーブル全体の読み込みを実行するのに必要なエクステント I/O の数。
ipagecrnt	float	ノンクラスタード・インデックスのリーフ・レベル全体の読み込みを実行する必要があるエクステント I/O の数。
drowcrnt	float	テーブル全体の読み込みを実行するのに必要なページ I/O の数。
oamapgcnt	int	テーブルの OAM ページの数 + テーブル情報を格納するアロケーション・ページの数。
extent0pgcnt	int	アロケーション・マップ・ページと同一のエクステント上のページ・カウント。
datarowsiz	float	データ・ローの平均サイズ。
leafrowsiz	float	ノンクラスタード・インデックスおよびクラスタード・インデックス・データオンリーロック・テーブルに対するリーフ・ローの平均サイズ。
status	int	内部システム・ステータス情報 (表 1-28 を参照)。
pljoindegree	int	ネスト・ループ・ジョイン操作で使用される並列度。pljoindegree は、ネスト・ループ・ジョインの内部テーブルにあたるテーブル (sysabstats がこのフィールドを持つテーブル) のスキャン並列度。
spare2	float	予約済み。
rslastoam	int	reorg reclaim_space コマンドまたは reorg compact コマンドによって最後に一時使用された OAM ページ。
rslastpage	int	reorg reclaim_space コマンドまたは reorg compact コマンドによって最後に一時使用されたデータ・ページまたはリーフ・ページ。
frlastoam	int	reorg forwarded_rows コマンドによって最後に一時使用された OAM ページ。

名前	データ型	説明
frlastpage	int	reorg forwarded_rows コマンドによって最後に一時使用されたデータ・ページ。
conopt_thld	smallint	同時実行性最適化スレッシュホールド。
plldegree	int16	データ操作言語 (DML) に対し、テーブルまたはインデックスで取り得る最大並列度。値 0 (ゼロ) は最大値が存在しないことを示す。この場合最大並列度はクエリ・プロセッサが設定する。
emptypgcnt	int	テーブルまたはインデックスに割り付けられたエクステント内の空のページ数。
spare4	float	予約済み。
partitionid	int	パーティション ID。
spare5	int	配置を揃えるための予備のフィールド。
statmoddate	datetime	ローが最後にディスクにフラッシュされた時刻。
unusedpgcnt	int	未使用のページ数。
oampagecnt	int	オブジェクト・アロケーション・マップにリストされているアロケーション・ページの数。

表 1-28 に、status カラムのビット表示をリストします。

表 1-28: sysindexes テーブルのステータス・ビット

10 進数	16 進数	ステータス
1	0x1	統計情報がアップグレードの結果である (update statistics ではない)。

インデックス

- id、indid、partitionid カラムに、ユニーク・クラスタード・インデックスが作成されています。

systhresholds

すべてのデータベース

説明 systhresholds には、データベースに定義されたスレッシュホールドごとに 1 つのローが含まれています。

カラム systhresholds のカラムは次のとおりです。

名前	データ型	説明
segment	smallint	空き領域があるかどうかについてモニタされているセグメント番号。
free_space	int	論理ページ単位のスレッシュホールドのサイズ。
status	smallint	ログ・セグメントのラストチャンス・スレッシュホールドに対しては 1、その他のすべてのスレッシュホールドに対しては 0。
proc_name	varchar(255)	segment で使用していない数が free_space を下回った場合に実行されるプロシージャ名。
suid	int null	スレッシュホールドを追加または最後に変更したユーザのサーバ・ユーザ ID。
currauths	varbinary(255) null	スレッシュホールドが追加されたか最後に修正されたとき、suid に対してアクティブな役割を示すビットマスク。スレッシュホールドを超えると、これらの役割から、スレッシュホールドが追加または最後に変更されてから無効になった役割を除いたものを使用して、proc_name が実行される。

currauth のアクティブな役割の特定 次の表に、currauth カラムに単独で、または組み合わせて指定できるビットマスクを示します。

10 進数	16 進数	説明
1	0x1	sa_role
2	0x2	sso_role
4	0x4	oper_role
8	0x8	sybase_ts_role
16	0x10	sybase_ts_role
32	0x20	navigator_ole
128	0x80	replication_role
256	0x100	dtm_tm_role
1024	0x400	ha_role
2048	0x800	mon_role
4096	0x1000	js_admin_role
16384	0x4000	messaging_role
32768	0x8000	web_services

Adaptive Server で `currauth` のビットマスク出力にどの役割 ID が関連付けられているかを判断するには、次の `select` 文を実行します。

```
1> select (c.number - 1) as role_id,role_name(c.number - 1) as role_name
2> from systhresholds ,master.dbo.spt_values c
3> where convert(tinyint,substring(isnull(currauth,0x1), c.low,1)) &
4> c.high != 0
5> and c.type = "P"
6> and c.number <= 1024
7> and c.number >0
8> and role_name(c.number - 1) is not null
9> go
```

Adaptive Server から、次のような出力が返されます。

role_id	role_name
0	sa_role
1	sso_role
2	oper_role
3	sybase_ts_role
4	navigator_role
7	dtm_tm_role
10	mon_role
11	js_admin_role
12	messaging_role
13	js_client_role

systemranges

master データベースのみ

説明 systemranges には、リソース制限をアクティブにする期間を制御するために Adaptive Server が使用する名前付き時間範囲が保管されます。

カラム systemranges のカラムは次のとおりです。

名前	データ型	説明
name	varchar(255)	時間範囲のユニークな名前。
id	smallint	時間範囲のユニークな識別子。1 は、“at all times” 制限を示す。
startday	tinyint	範囲の最初にあたる曜日 (1 ~ 7)。月曜日 = 1、日曜日 = 7。
endday	tinyint	範囲の最後にあたる曜日 (1 ~ 7)。月曜日 = 1、日曜日 = 7。
starttime	varchar(10)	範囲の開始時刻。
endtime	varchar(10)	範囲の終了時刻。

インデックス

- id カラムに、クラスタード・インデックスが作成されています。

systransactions

master データベースのみ

説明 systransactions には Adaptive Server のトランザクションについての情報が含まれていますが、このテーブルは、通常のテーブルではありません。テーブルの一部は、ユーザによるクエリが実行されたときに動的に構築されますが、その他の部分は master データベースに格納されます。systransactions の動的に構築されたカラムの更新はできません。

カラム systransactions のカラムは次のとおりです。

名前	データ型	説明
xactkey	binary(14)	ユニーク Adaptive Server トランザクション・キー。
starttime	datetime	トランザクションが開始した日付。
failover	int	トランザクションのフェールオーバー・ステータスを示す値 (表 1-29 を参照)。
type	int	トランザクションのタイプを示す値 (表 1-30 を参照)。
coordinator	int	コーディネーションの方法またはプロトコルを示す値 (表 1-31 を参照)。
state	int	トランザクションの現在のステータスを示す値 (表 1-32 を参照)。
connection	int	接続ステータスを示す値 (表 1-33 を参照)。
status	int	内部トランザクション・ステータス・フラグ。
status2	int	追加の内部トランザクション・ステータス・フラグ。
spid	smallint int (Cluster Edition)	サーバ・プロセス ID、またはプロセスが接続されていない場合は 0。
masterdbid	smallint	トランザクションのデータベースを起動中。
loid	int	ロック所有者 ID。
namelen	smallint	xactname の長さ。
xactname	varchar(255) null	トランザクション名または XID。
srvname	varchar(30) null	リモート・サーバの名前 (ローカル・サーバの場合は null)。
nodeid	tinyint null	今後のために予約済み (クラスタ環境では使用不可)。
instanceid	tinyint	インスタンスの ID (クラスタ環境のみで使用可能)。

注意 Cluster Edition ではデータ型にこの変更が加えられたため、監査テーブルをアップグレードする前にアーカイブおよびトランケートすることを強くおすすめします。それによって、sybsecurity データベース内の領域不足によるアップグレードの失敗を低減できます。

表 1-29: *systransactions failover* カラム値

<i>failover</i> 値	フェールオーバー・ステータス
0	Resident Tx
1	Failed-over Tx
2	Tx by Failover-Conn

表 1-30: *systransactions type* カラム値

<i>type</i> 値	トランザクション・タイプ
1	Local
3	External
98	Remote
99	Dtx_State

表 1-31: *systransactions coordinator* カラム値

<i>coordinator</i> 値	コーディネーションの方法またはプロトコル
0	なし
1	Syb2PC
2	ASTC
3	XA
4	DTC

表 1-32: *systransactions state* カラム値

<i>state</i> 値	トランザクションのステータス
1	Begun
2	Done Command
3	Done
4	Prepared
5	In Command
6	In Abort Cmd
7	Committed
8	In Post Commit
9	In Abort Tran
10	In Abort Savept
65537	Begun-Detached
65538	Done Cmd-Detached
65539	Done-Detached
65540	Prepared-Detached
65548	Heur Committed
65549	Heur Rolledback

表 1-33: *systransactions connection* カラム値

<i>connection</i> 値	接続ステータス
1	Attached
2	Detached

systypes

すべてのデータベース

説明 `systypes` には、システムが提供するデータ型およびユーザ定義データ型ごとに 1 つのローが含まれています。ドメイン (ルールで定義される) とデフォルトが存在する場合は、それらも含まれています。

システムが提供するデータ型を示すローは、変更できません。

カラム `systypes` のカラムは次のとおりです。

名前	データ型	説明
uid	int	データ型を作成したユーザ ID。
usertype	smallint	ユーザ・タイプ ID。
variable	bit	データ型が可変長の場合は 1、それ以外は 0。
allownulls	bit	データ型に null が使用できるかどうかを示す。
type	tinyint	物理記憶データ型。
length	int	データ型の物理長。
tdefault	int	データ型のデフォルトを生成するシステム・プロシージャの ID。
domain	int	データ型の整合性チェックを含むシステム・プロシージャの ID。
name	varchar(255)	データ型名。
printfmt	varchar(255) null	予約済み。
prec	tinyint null	有効桁数。
scale	tinyint null	小数点以下の桁数。
ident	tinyint null	1 はカラムに IDENTITY プロパティがあることを示す。0 は IDENTITY プロパティがないことを示す。
hierarchy	tinyint null	混合モード算術式でのデータ型の優先度。
xtypeid	int null	内部クラス ID。
xdbid	int null	クラスがインストールされている dbid。 <ul style="list-style-type: none"> • -1 = システム・データベース • -2 = 現在のデータベース
accessrule	int null	<code>sysprocedures</code> におけるアクセス・ルールのオブジェクト ID。

表 1-34 に、システムが提供する各データ型の `name`、`hierarchy`、`type` (必ずしもユニークではない)、および `usertype` (ユニーク) をリストします。データ型は階層順に並んでいます。混合モードの算術では、`hierarchy` (階層) の低いデータ型から優先されます。

表 1-34: データ型名、階層、タイプ、ユーザ・タイプ

名前	階層	タイプ	ユーザ・タイプ
floatn	1	109	14
float	2	62	8
datetimn	3	111	15
datetime	4	61	12
real	5	59	23
numericn	6	108	28
numeric	7	63	10
decimaln	8	106	27
decimal	9	55	26
moneyn	10	110	17
money	11	60	11
smallmoney	12	122	21
smalldatetime	13	58	22
intn	14	38	13
uintn	15	68	47
bigint	16	191	43
ubigint	17	67	46
int	18	56	7
uint	19	66	45
smallint	20	52	6
usmallint	21	65	44
tinyint	22	48	5
bit	23	50	16
univarchar	24	155	35
unichar	25	135	34
unitext	26	174	36
varchar	27	39	2
sysname	27	39	18
nvarchar	27	39	25
longsysname	27	39	42
char	28	47	1
nchar	28	47	24
varbinary	29	37	4
timestamp	29	37	80
binary	30	45	3
text	31	35	19
image	32	34	20
date	33	49	37
time	34	51	38

名前	階層	タイプ	ユーザ・タイプ
daten	35	123	39
timen	36	147	40
extended type	99	36	-1

インデックス

- **name** カラムに、ユニーク・クラスタード・インデックスが作成されています。
- **usertype** カラムに、ユニーク・ノンクラスタード・インデックスが作成されています。

sysusages

master データベースのみ

説明

sysusages には、データベースに割り当てられたディスク割り付けの集まりごとに 1 つのローが含まれています。各データベースは、指定された数のデータベース (論理) ページ番号を持っています。

create database コマンドは、**sysdevices** と **sysusages** をチェックして使用できるディスク割り付けの集まりを検出します。1 つまたは複数の連続するディスク割り付けの集まりがデータベースに割り当てられると、**sysusages** にマッピングが記録されます。

sysusages の詳細については、『システム管理ガイド』の「第 21 章 ユーザ・データベースの作成と管理」の「領域の割り付けを管理するシステム・テーブル」を参照してください。

注意 Adaptive Server バージョン 15.0 では、デバイス ID 番号は **vdevno** カラムに格納されており、**vstart** カラムには含まれていません。したがって、以前のスキーマに基づいてデバイス ID 番号を判断するスクリプトとストアド・プロシージャは、修正が必要な場合があります。

カラム

sysusages のカラムは次のとおりです。

名前	データ型	説明
dbid	smallint	データベース ID。
segmap	int	可能なセグメント割り当てのビットマップ。
lstart	int	最初のデータベース (論理) ページの番号。
size	int	連続したデータベース (論理) ページの数。
vstart	int	仮想ページの開始番号。
pad	smallint null	未使用。
unreservedpgs	int null	割り当てられたエクステンション部分以外の空き領域。
crdate	datetime null	作成した日付。
vdevno	int	デバイス ID 番号。

インデックス

- **dbid**、**lstart** カラムに、ユニーク・クラスタード・インデックスが作成されています。
- **vdevno**、**vstart** カラムに、ユニーク・ノンクラスタード・インデックスが作成されています。

sysusermessages

すべてのデータベース

説明 `sysusermessages` には、Adaptive Server によって返されるユーザ定義メッセージごとに 1 つのローが含まれています。

カラム `sysusermessages` のカラムは次のとおりです。

名前	データ型	説明
error	varchar(1024)	ユニークなエラー番号。20,000 以上である。
uid	int	メッセージ作成者のサーバのユーザ ID (<code>suser_id</code>)。
description	varchar(1024)	パラメータ用のオプションのプレースホルダを持つユーザ定義のメッセージ。
langid	smallint null	メッセージの言語 ID。us_english では null。
dlevel	smallint null	with_log ビットを保管する。このビットは、メッセージのログを取るために適切なルーチンを呼び出すのに使用される。

インデックス

- error カラムに、クラスタード・インデックスが作成されています。
- error、langid カラムに、ユニーク・ノンクラスタード・インデックスが作成されています。

sysusers

すべてのデータベース

説明 `sysusers` には、データベースで許可されたユーザごとに 1 つのロー、およびグループまたは役割ごとに 1 つのローが含まれています。

カラム `sysusers` のカラムは次のとおりです。

名前	データ型	説明
suid	int	syslogins からコピーされるサーバのユーザ ID。
uid	int	ユーザ ID。データベース内ではユニークで、パーミッションの付与および取り消しに使用される。ユーザ ID が 1 の場合、“dbo” を示す。
gid	int	ユーザが所属するグループ ID。uid = gid の場合、このエントリがグループを定義する。ユーザ ID (uid) に負の値が使用されることがある。sysusers でグループや役割に関連付けられている suid はすべて、-2 (INVALID_SUID) に設定される。
name	sysname	ユーザ名またはグループ名。このデータベース内ではユニーク。
environ	varchar(255) null	予約済み。

Adaptive Server の配布メディアでは、`master..sysusers` にはいくつかの初期ユーザが含まれています。そのうちの 1 つが “dbo” で、suid は 1、uid は 1 に設定されています。2 つ目は “guest” で、suid は -1、uid は 2 に設定されています。3 つ目は “public” で、suid は -2、uid は 0 に設定されています。さらに、システム標準の役割とユーザ定義の役割の両方が `sysusers` 内にリストされます。

“guest” ユーザは、`sysusers` に明示的にリストされていないユーザに対して、制限されたパーミッションのセットを使用してデータベースにアクセスできるメカニズムを提供します。`master` データベースの “guest” のエントリは、Adaptive Server にアカウントを持つユーザ (つまり `syslogins` にエントリを持つユーザ) であれば、`master` データベースにアクセスできることを意味します。

“public” ユーザはすべてのユーザを意味します。`public` キーワードは、`grant` コマンドまたは `revoke` コマンドと一緒に使用され、パーミッションを付与したり取り消したりする対象がすべてのユーザの場合に使用します。

インデックス

- `suid` カラムに、ユニーク・クラスタード・インデックスが作成されています。
- `name` カラムに、ユニーク・ノンクラスタード・インデックスが作成されています。
- `uid` カラムに、ユニーク・ノンクラスタード・インデックスが作成されています。

sysxtypes

すべてのデータベース

説明 sysxtypes には、拡張機能 Java-SQL データ型ごとに 1 つのローが含まれています。

Java-SQL クラスとデータ型の詳細については、『Adaptive Server Enterprise における Java』を参照してください。

カラム sysxtypes のカラムは次のとおりです。

名前	データ型	説明
xtid	int	拡張型に対するシステム生成 ID。
xtstatus	int	内部ステータス情報。未使用。
xtmetatype	int	未使用。
xtcontainer	int	クラスを含んでいる JAR ファイルの ID。NULL の場合もある。
xtname	varchar(255) null	拡張型の名前。
xtsource	text null	拡張型のソース・コード。未使用。
xtbinaryinrow	varbinary(255) null	拡張型のオブジェクト・コード。Java クラスの場合、クラス・ファイルを含む。データは、最大 255 バイトのロー状態で格納されています。
xtbinaryoffrow	image	拡張型のオブジェクト・コード。Java クラスの場合、クラス・ファイルを含む。データは、イメージ・カラムとして、ローでない状態で格納される。

インデックス

- **xtid** カラムに、ユニーク・クラスタード・インデックスが作成されています。
- **xtname** カラムに、ユニーク・ノンクラスタード・インデックスが作成されています。

すべてのデータベースに含まれる標準のシステム・テーブルに加えて、dbcc 管理データベース dbccdb には、dbcc checkstorage への入力とそこから出力を定義する 7 つのテーブルが含まれます。また、このデータベースには、少なくとも 2 つの作業領域も含まれます。

トピック名	Page
dbccdb 作業領域	101
dbccdb ログ	103

dbccdb 作業領域

作業領域とは、dbcc checkstorage オペレーションの中間結果を格納する dbccdb 内の特殊テーブルをいいます。作業領域は、次の点でワーク・テーブルと異なります。

- I/O パフォーマンスを向上させるために事前に継続して割り付けられる
- 永続的である
- tempdb データベースに常駐しない

dbccdb を作成すると、2 つの作業領域が自動的に作成されます。これらは、次のように事前に割り付けられます。

- スキャン作業領域 – ターゲット・データベースの各ページのローが含まれます。割り付けは、データベース・サイズのおよそ 1 パーセントになります。各ローは、単一の binary(18) カラムで構成されます。
- テキスト作業領域 – text または image カラムを含むターゲット・データベース内の各テーブルのローが含まれます。このテーブルのサイズは、ターゲット・データベースの設計によって異なりますが、通常は、スキャン作業領域よりもはるかに小さくなります。各ローは、単一の binary(22) カラムで構成されます。

どちらかの割り付けが、dbcc checkstorage で必要なサイズよりも大きい場合、オペレーションでは必要なだけが使用されます。割り付けは変更されません。テキスト作業領域の割り付けが小さすぎる場合、dbcc checkstorage はこのことをレポートし、新しいサイズをすすめて、検査を続けます。ただし、一部のテキスト・チェーンは検査されません。スキャン作業領域の割り付けが小さすぎる場合、dbcc checkstorage オペレーションはすぐに失敗します。

少なくとも1つのスキャン作業領域と1つのテキスト作業領域が必要ですが、任意の数の作業領域を作成できます。使用中の作業領域はロックされて、所定の時刻に1つの **dbcc checkstorage** オペレーションだけがそれらを使用できるようになります。同時 **dbcc checkstorage** オペレーションは、それぞれに独立したスキャン作業領域とテキスト作業領域を提供することによって実行できます。

作業領域の作成に関する詳細については、『システム管理ガイド』と『Adaptive Server リファレンス・マニュアル』を参照してください。

理想としては、**dbcc checkstorage** によってのみ作業領域にアクセスするべきですが、これは稼働条件ではありません。**dbcc checkstorage** は、使用している作業領域を排他ロックし、作業領域の内容は **dbcc checkstorage** を実行するたびに再生成されます。作業領域にはセキュア・データは含まれません。

注意 作業領域の内容には SQL でアクセスできますが、バイナリ値は解釈されません。SQL によるアクセスでは、異なる **dbcc** 検査からのデータが混ざって返される場合があります。これらのテーブル内にローが存在しても、有効なデータが含まれるとは限りません。**dbcc** は、実行中にのみ有効なローを追跡します。この情報は、オペレーションが終了すると失われます。

dbccdb のほとんどの更新アクティビティは、テキスト作業領域およびスキャン作業領域で実行されます。作業領域は事前に割り付けられており、これらの作業領域を使用できるのは常に1つの **dbcc checkstorage** だけであるため、作業領域が破壊される可能性は、多くのユーザ・テーブルよりも低くなります。作業領域が破壊されると、**dbcc checkstorage** オペレーションが失敗するか、誤動作が生じる可能性があります。この場合は、破壊された作業領域を削除して再作成してください。

異なる作業領域を使用する複数のデータベースの検査は同時に進めることができますが、ディスク・スループットの競合が生じるため、各オペレーションのパフォーマンスは低下するおそれがあります。

作業領域を削除するには、**dbccdb** で次のように入力します。

```
drop table workspace_name
```

dbccdb ログ

各 dbcc checkstorage オペレーションの結果は、dbccdb ログに記録されます。テキスト作業領域およびスキャン作業領域への更新は、ここには記録されません。

dbccdb ログは、テーブルへの更新を処理するようにサイズ指定する必要があります。ログの稼働条件は、ターゲット・データベース内のテーブルとインデックスの数に関連します。これは、ターゲット・データベースのサイズに関係ありません。

ログの稼働条件とリカバリ時間を最小限に抑えるには、dbccdb に truncate log on checkpoint オプションを付けて使用してください。

dbcc_config

説明

dbcc_config テーブルは、現在実行中の、または最後に完了した dbcc checkstorage オペレーションを記述します。このテーブルは、次のものを定義します。

- dbcc checkstorage オペレーション専用リソースのロケーション
- dbcc checkstorage オペレーションでのリソース使用率の制限値

カラム

dbcc_config のカラムは次のとおりです。

カラム名	データ型	説明
dbid	smallint	sysindatabases のローの dbid に対応する。
type_code	int	dbcc_types テーブルのローの type_code に対応する。有効な値は 1 ~ 9。
value	int null	type_code によって示される項目の値を示す。stringvalue の値が null ではない場合のみ、null を指定できる。
stringvalue	varchar(255) null	type_code によって示される項目の値を示す。value の値が null ではない場合のみ、null を指定できる。

プライマリ・キー

dbid と type_code の組み合わせです。

参照

dbcc_config の初期化と更新については、『システム管理ガイド』を参照してください。

dbcc_counters

説明 dbcc_counters テーブルには、dbcc checkstorage によって実行された分析の結果が格納されます。カウンタは、データベース、テーブル、インデックス、パーティション、デバイス、および dbcc の呼び出しごとに維持されます。

カラム dbcc_counters のカラムは次のとおりです。

カラム名	データ型	説明
dbid	smallint	ターゲット・データベースを示す。
id	int	テーブルを示す。この値は、sysindexes と sysobjects から抽出される。
indid	smallint	インデックスを示す。この値は、sysindexes から抽出される。
partitionid	int	定義されたオブジェクト・ページの結び付きを示す。この値は、sysindexes と syspartitions から抽出される。
devid	int	ディスク・デバイスを示す。この値は、sysdevices から抽出される。
opid	smallint	実行された dbcc オペレーションを示す。
type_code	int	dbcc_types テーブルのローの type_code カラムに対応する。有効な値は 5000 ~ 5024。
value	real null	dbcc_types に記述された特定の type_code の適切な type_name に対応する。

プライマリ・キー dbid、id、indid、partitionid、devid、opid、および type_code の組み合わせです。

dbcc_exclusions

説明 dbcc_exclusions テーブルは、checkverify による処理と sp_dbcc_faultreport によるフォールト報告から除外される、フォールト、テーブル、フォールトとテーブルの組み合わせを格納します。

カラム dbcc_exclusionss のカラムは次のとおりです。

カラム名	データ型	説明
dbid	smallint	ターゲット・データベースを示す。
type	tinyint	除外タイプ・コード。有効な値は次のとおり。 <ul style="list-style-type: none"> • 1 – フォールト • 2 – テーブル • 3 – 組み合わせ
fault_type	int null	type が 1 (フォールト) または 3 (組み合わせ) の場合に、除外するフォールト・タイプ。詳細については、「 dbcc のタイプ 」(111 ページ) を参照。
table_name	varchar(30) null	type が 2 (テーブル) または 3 (組み合わせ) の場合に、除外するテーブル名。詳細については、「 dbcc のタイプ 」(111 ページ) を参照。

プライマリ・キー dbid、fault_type、table_name の組み合わせです。

dbcc_fault_params

説明 dbcc_fault_params テーブルには、dbcc_faults テーブルに入力されたフォールトに関する補足説明が含まれます。

カラム dbcc_fault_params のカラムは次のとおりです。

カラム名	データ型	説明
dbid	smallint	ターゲット・データベースを示す。
opid	smallint	実行された dbcc オペレーションを示す。
faultid	int	フォールト ID を示す。
type_code	int	“value” カラムによって提供される値の解釈を定義する。有効な値は 1000 ~ 1009。これらの値は、 dbcc_types に示されている。
intvalue	int null	整数値を示す。
realvalue	real null	実数値を示す。
binaryvalue	varbinary(255) null	バイナリ値を示す。
stringvalue	varchar(255) null	文字列値を示す。
datevalue	datetime null	日付値を示す。

各 “value” カラム (intvalue、realvalue、binaryvalue、stringvalue、および datevalue) には、null 値を指定できます。ただし、少なくとも 1 つのカラムには null 以外の値を指定する必要があります。これらのカラムのうち複数のカラムに null 以外の値が含まれる場合、それらのカラムは同じ値を異なる表現で示します。

プライマリ・キー dbid、opid、faultid、type_code の組み合わせです。

dbcc_faults

説明 dbcc_faults テーブルには、dbcc checkstorage によって検出される各フォールトの説明が示されます。

カラム dbcc_faults のカラムは次のとおりです。

カラム名	データ型	説明
dbid	smallint	ターゲット・データベースを示す。
id	smallint	テーブルを示す。この値は、sysindexes と sysobjects から抽出される。
indid	smallint	インデックスを示す。この値は、sysindexes から抽出される。
partitionid	int	パーティションを示す。この値は、sysindexes と syspartitions から抽出される。カウンタはページ範囲ごとに維持されるため、「パーティション」とは、実際のオブジェクト・ページのチェーンではなく、定義されたオブジェクト・ページの結び付きを指す。
devid	int	ディスク・デバイスを示す。この値は、sysdevices から抽出される。
opid	smallint	実行された dbcc オペレーションを示す。
faultid	int	オペレーションで記録された各フォールトに割り当てられたユニークなシーケンス番号を指定する。
type_code	int	フォールトのタイプを示す。有効な値は 100000 ~ 100032。これらの値は、 表 2-1 (111 ページ) に示されている。
status	int	<p>フォールトを分類する。有効な値は次のとおり。</p> <ul style="list-style-type: none"> • 0 – ソフト・フォールト、おそらく一時的 • 1 – ハード・フォールト • 2 – 一時的であることが実証されたソフト・フォールト • 3 – ハード・フォールトにアップグレードされたソフト・フォールト • 5 – 修復されたハード・フォールト • 7 – アップグレード後に修復されたハード・フォールト • 9 – 修復不可能なハード・フォールト • 11 – ハード・フォールトにアップグレードされた、修復不可能なソフト・フォールト • 16 – オブジェクトが削除されたソフト・フォールト (アクセス不可) • 17 – オブジェクトが削除されたハード・フォールト (アクセス不可) • 18 – オブジェクトが削除された一時ソフト・フォールト (アクセス不可) • 19 – ハード・フォールトにアップグレードされ、オブジェクトが削除されたソフト・フォールト (アクセス不可) <p>詳細については、『システム管理ガイド』を参照してください。</p>

プライマリ・キー dbid, id, indid, partitionid, devid, opid, faultid, type_code の組み合わせです。

dbcc_operation_log

説明 dbcc_operation_log テーブルには、dbcc checkstorage オペレーションの使用状況が記録されます。

カラム dbcc_operaiton_log のカラムは次のとおりです。

カラム名	データ型	説明
dbid	smallint	ターゲット・データベースを示す。
opid	smallint	dbcc checkstorage オペレーションのシーケンス番号を示す。opid は、自動的に増分する番号であり、dbid と finish の組ごとにユニークである。
optype	smallint	次に optype に有効な値を示す。 • 2 = checkstorage
suid	int	コマンドを実行するユーザを示す。
start	datetime	オペレーションがいつ開始したかを示す。
finish	datetime null	オペレーションがいつ終了したかを示す。
seq	smallint null	checkverify オペレーションのシーケンス番号。
id	int null	checkverify オペレーションのオブジェクト ID (使用された場合)。
maxseq	smallint null	checkstorage オペレーションに対する checkverify で使用された最大のシーケンス。

計算結果は、dbcc_operation_results テーブルに記録されます。

プライマリ・キー dbid、opid、optype の組み合わせです。

dbcc_operation_results

説明 dbcc_operation_results テーブルには、dbcc_operation_log テーブルに記録されたオペレーションに関する補足説明が示されます。

カラム dbcc_operation_results のカラムは次のとおりです。

カラム名	データ型	説明
dbid	smallint	ターゲット・データベースを示す。
opid	smallint	dbcc オペレーション ID を示す。
optype	smallint	dbcc オペレーション・タイプを示す。
type_code	int	dbcc オペレーション・タイプを定義する。有効な値は 1000 ~ 1007。これらの値は、 表 2-1 (111 ページ) に示されている。
intvalue	int null	整数値を示す。
realvalue	real null	実数値を示す。
binaryvalue	varbinary(255) null	バイナリ値を示す。
stringvalue	varchar(255) null	文字列値を示す。
datevalue	datetime null	日付値を示す。
seq	smallint null	checkverify オペレーションのシーケンス番号。

各“value”カラム (intvalue、realvalue、binaryvalue、stringvalue、および datevalue) には、null 値を指定できます。ただし、少なくとも 1 つのカラムには null 以外の値を指定する必要があります。これらのカラムのうち複数のカラムに null 以外の値が含まれる場合、それらのカラムは同じ値を異なる表現で示します。

dbcc checkstorage オペレーションの結果には、次の数が示されます。

- 検出されたハード・フォールト
- 検出されたソフト・フォールト
- ハード・エラーが原因で停止したオペレーション

プライマリ・キー dbid、opid、optype、type_code の組み合わせです。

dbcc_types

説明 dbcc_types テーブルには、dbcc checkstorage によって使用されるデータ型の定義が示されます。このテーブルは、実際には dbcc ストアド・プロシージャでは使用されません。このテーブルは、dbccdb 内にある他のテーブルの使用を簡単にして、データ型のセマンティックを記述するために用意されています。オペレーション設定のタイプ・コード、レポートされた分析データ、フォールトの分類、およびフォールト・レポート・パラメータが含まれます。レポートを生成するための独自のストアド・プロシージャを作成する場合は、type_name カラムにリストされた値をレポート見出しとして使用できます。

カラム dbcc_types のカラムは次のとおりです。

注意 dbcc_types への今後の追加を可能にするために、ここでは一部の type_code 番号が使用されていません。

表 2-1: dbcc のタイプ

type_code	type_name	説明
1	max worker processes	オプション - 適用できるワーカー・プロセスの最大数を示す。使用される同時処理の最大レベルも示す。最大値は 1。
2	dbcc named cache	dbcc checkstorage によって使用されるキャッシュのサイズ(キロバイト数)とそのキャッシュの名前を示す。
3	scan workspace	データベース・スキャンで使用される作業領域の ID と名前を示す。
4	text workspace	text カラムに使用される作業領域の ID と名前を示す。
5	operation sequence number	最も新しく開始された dbcc オペレーションを示す番号を示す。
6	database name	sysdatabases にデータベースの名前を示す。
7	OAM count threshold	オブジェクト・アロケーション・マップ・カウントがエラーと見なされる時点のパーセンテージを示す。
8	IO error abort	dbcc がディスク上でページ検査を停止する時点の、そのディスクの I/O エラー数を示す。
9	linkage error abort	dbcc がオブジェクトのページ・チェーンを停止する時点の、リンケージ・エラー数を示す。一部のページ・チェーン破壊では、他のページ・チェーン破壊よりも少ない数のリンケージ・エラーが発生した時点で、検査を停止しなければならない場合がある。
10	enable automatic workspace expansion	作業領域の予想サイズが実際の作業領域のサイズを超える場合に、作業領域の自動拡張を有効または無効にするフラグ。
1000	hard fault count	一貫性の検査中に検出された永続的な矛盾(ハード・フォールト)の数を示す。
1001	soft fault count	一貫性の検査中に検出された疑わしい条件(ソフト・フォールト)の数を示す。
1002	checks aborted count	一貫性の検査中に停止されたリンケージ検査の数を示す。
1007	text column count	一貫性の検査中に検出された null 以外の text カラムおよび image カラムの数を示す。

type_code	type_name	説明
5000	bytes data	検査対象パーティションに格納されるユーザ・データの量をバイト数で示す。
5001	bytes used	検査対象パーティション内のデータを記録するために使用される記憶領域の量をバイト数で示す。bytes used と bytes data の差は、データの格納またはインデックス作成に必要なオーバーヘッドの量を示す。
5002	pages used	オブジェクトを保持するために実際に使用される、検査対象オブジェクトにリンクされたページ数を示す。
5003	pages reserved	検査対象オブジェクトに予約されているが、そのオブジェクトで使用するために割り付けられていないページ数を示す。(8 * extents used) と (pages used + pages reserved) の差は、コミットされていない割り付け解除と不正に割り付けられたページ数の合計を示す。
5004	pages overhead	オブジェクト・アロケーション・マップ・ページまたはインデックス統計などのオーバーヘッド関数に使用されるページ数を示す。
5005	extents used	検査対象パーティション内のオブジェクトに割り付けられたエクステンント数を示す。オブジェクト 99 (アロケーション・ページ) の場合、この値は、有効なオブジェクトに割り付けられていないエクステンントの数を示す。オブジェクト 99 には、他のオブジェクトに割り付けられていない記憶領域が含まれる。
5006	count	検査対象オブジェクトの一部にあるいずれかのページで検出されたコンポーネント項目 (ローまたはキー) の数を示す。
5007	max count	検査対象オブジェクトの一部にあるいずれかのページで検出されたコンポーネント項目の最大数を示す。
5008	max size	検査対象オブジェクトの一部にあるいずれかのページで検出されたコンポーネント項目の最大サイズを示す。
5009	max level	インデックス内のレベルの最大数を示す。このデータ型は、テーブルには適用できない。
5010	pages misallocated	オブジェクトに割り付けられたが、正しく初期化されないページ数を示す。これは、フォールト・カウンタである。
5011	io errors	検出された I/O エラーの数を示す。このデータ型は、フォールト・カウンタである。
5012	page format errors	レポートされたページ・フォーマット・エラーの数を示す。このデータ型は、フォールト・カウンタである。
5013	pages not allocated	チェーンによってオブジェクトにリンクされたが、割り付けられていないページ数を示す。このデータ型は、フォールト・カウンタである。
5014	pages not referenced	オブジェクトに割り付けられたが、そのチェーンを介して到達していないページ数を示す。このデータ型は、フォールト・カウンタである。
5015	overflow pages	検出されたオーバフロー・ページの数を示す。このデータ型は、クラスタード・インデックスにのみ適用できる。
5016	page gaps	次のページに昇順でリンクされていないページ数を示す。この数字は、テーブル断片化の量を示す。
5017	page extent crosses	各ページのエクステンントの外部にあるページにリンクされたページ数を示す。page extent crosses の数は pages used または extents used に比例して増えるため、大容量 I/O バッファの効率は低下する。

<i>type_code</i>	<i>type_name</i>	説明
5018	page extent gaps	次のエクステントが昇順による次のエクステントではないページ・エクステント・クラスの数を示す。フル・スキャンでの I/O パフォーマンスは、page extent gaps の値が最も小さいときに最大になる。シークまたはフル・ディスク・ローテーションは、各ギャップで発生する可能性が高くなる。
5019	ws buffer crosses	dbcc checkstorage オペレーション中に各ページの作業領域バッファ・キャッシュの外部にリンクされたページの数を示す。この情報は、キャッシュのサイズ指定に使用して、リソースを無駄にすることなく高いパフォーマンスを得ることができる。
5020	deleted rows	オブジェクトで削除されたローの数。
5021	forwarded rows	オブジェクトで転送されたローの数。
5022	empty pages	割り付けられた、データのないページ数。
5023	pages with garbage	ガーベジ・コレクションを利用できるページ。
5024	non-contiguous free space	不連続空き領域のバイト数。
10000	page id	フォールト検出時に検査されていたページのデータベース内のローケーションを示す。ローカライズされたフォールトすべてにこのパラメータが含まれる。
10001	page header	フォールト検出時に検査されていたページのヘッダの 16 進表現を示す。この情報は、ソフト・フォールトの評価と検査以降にページが更新されているかどうかの判断に役立つ。後続のゼロはサーバによってトランケートされる。
10002	text column id	フォールトのあったテキスト・チェーンへの参照のページ、ロー、およびカラムを示す 8 バイトの 16 進値を示す。後続のゼロはサーバによってトランケートされる。
10003	object id	object id (テーブル)、該当する場合は partition id (テーブルのパーティション)、および検査対象のページまたは割り付けの index id (インデックス) を示す 9 バイトの 16 進値を示す。 たとえば、ページが T1 のチェーンから到達したために、テーブル T1 に属するものと予期されるが、実際にはテーブル T2 に割り付けられる場合は、T1 の object id が記録されて、T2 の object id expected が記録される。後続のゼロはサーバによってトランケートされる。
10007	page id expected	予期されるページ ID と実際に検出されたページ ID の間に違いがある場合は、リンク・ページに予期されるページ ID を示す。 たとえば、P1 から P2 にチェーンをたどるとき、前方に進んでから後方に戻る場合、P1 は P2 の後にくるものと予期される。page id expected の値は P1 であり、page id の値は P2 になる。P3 の実際の値が検出される場合は、この値が page id actual として記録される。
10008	page id actual	実際に検出されたページ ID と予期されるページ ID との間に違いがある場合、この値は、検出された実際のページ ID を示す (type_code 10007 も参照)。 たとえば、P1 から P2 にチェーンをたどるとき、前方に進んでから後方に戻る場合、P1 は P2 の後にあるものとみなされる。page id expected の値は P1 であり、page id の値は P2 になる。P3 の実際の値が検出される場合は、この値が page id actual として記録される。

type_code	type_name	説明
10009	object id expected	<p>予期されるオブジェクト ID (テーブル)、もしあればパーティション ID (テーブルのパーティション)、および検査対象のページまたは割り付けのインデックス ID (インデックス) を示す 9 バイトの 16 進値を示す。</p> <p>たとえば、ページが T1 のチェーンから到達したために、テーブル T1 に属するものと予期されるが、実際にはテーブル T2 に割り付けられる場合は、T1 の object id が記録されて、T2 の object id expected が記録される。後続のゼロはサーバによってトランケートされる。</p>
10010	data-only locked data page header	フォールトが発生したページの 44 バイトのページ・ヘッダを示す。
10011	data-only locked b-tree leaf page header	フォールトが発生したページの 44 バイトのページ・ヘッダを示す。
10012	data-only locked b-tree header	フォールトが発生したページの 44 バイトのページ・ヘッダを示す。
20001	rerun checkstorage reco	checkstorage を再実行する。
20002	indexalloc reco	dbcc indexalloc を fix オプション付きで実行する。
20003	tablealloc reco	dbcc tablealloc を fix オプション付きで実行する。
20004	checktable fix_spacebits reco	dbcc tablealloc を fix_spacebits オプション付きで実行する。
20005	checktable reco	dbcc checktable を実行する。
20006	reorg reco	reorg コマンドを実行する。
20007	no action reco	このフォールトは無害であり、アクションは不要。
30000	drop object reco	オブジェクトを削除して再作成する。
30001	bulk copy reco	バルク・コピーを使ってデータをコピー・アウトおよびコピー・インする。
40000	check logs for hardware failure reco	オペレーティング・システムのログをチェックし、Sybase デバイスを含むディスクに関してレポートされたハードウェアの問題をすべて修正する。
40001	checkalloc reco	dbcc checkalloc を fix オプション付きで実行する。
40002	reload db reco	クリーン・バックアップからデータベースを再ロードする。
100000	IO error	指定されたページの一部をデバイスから取り出せなかったことを示す。通常、オペレーティング・システムまたはハードウェアの失敗が原因で生じる。
100001	page id error	ページに記録された識別 ID (ページ番号) が無効であることを示す。これは、間違ったディスク・ロケーションに対するページの書き込みまたはそこからのページの読み取り、ページの書き込み前または書き込み時の破壊、あるいは後続の初期化が行われないページの割り付けが原因で生じる。
100002	page free offset error	ページのデータの終了が無効であることを示す。このイベントは、このページでの挿入と更新に影響を及ぼす。また、このページ上のデータに対する一部のアクセスにも影響する。

<i>type_code</i>	<i>type_name</i>	説明
100003	page object id error	ページが、予期されたもの以外のテーブルに割り付けられているように見えることを示す。永続的なフォールトである場合は、次のいずれかが原因として考えられる。 <ul style="list-style-type: none"> 不正なページ割り付け。次の割り付けでこのページが事実上失われる可能性がある。 ページ・チェーンの破壊。破壊されたチェーン内のデータにアクセスできなくなる可能性がある。
100004	timestamp error	ページにデータベースのタイムスタンプよりも後のタイムスタンプがあることを示す。このエラーが原因で、このページに変更が加えられたときにリカバリできなくなる可能性がある。
100005	wrong dbid error	データベース ID <code>dbid</code> がデータベース・アロケーション・ページに格納されていることを示す。この ID が不正だと、アロケーション・ページは破壊されて、指定された割り付けはすべて疑わしくなる。
100006	wrong object error	ページの割り付けに矛盾があることを示す。ページが、1つのテーブルまたはインデックスに属しているように見えるが、アロケーション・ページ内の別のテーブルまたはインデックスに割り付けられているものとして記録される。このエラーは、割り付けが矛盾しているが結果は同様であるという点で、 <code>page object id error</code> と異なる。
100007	extent id error	<code>dbcc checkstorage</code> に認識されないテーブルまたはインデックスへの割り付けが検出されたことを示す。通常この結果、割り付けられた記憶領域が使用できなくなる。
100008	fixed format error	ページが、単一固定長のローだけを含むということを誤って示すことを表す。 <code>dbcc checkstorage</code> によってこのエラーがレポートされる。 <code>dbcc checktable</code> はこの状態をレポートしないが、修復する。
100009	row format error	ページ上の少なくとも1つのローが不正にフォーマットされたことを示す。このエラーが原因で、このページ上の一部またはすべてのデータにアクセスできなくなる可能性がある。
100010	row offset error	ページ上の少なくとも1つのローが予期されるページ・オフセットにならないことを示す。このエラーが原因で、このページ上の一部またはすべてのデータにアクセスできなくなる可能性がある。
100011	text pointer error	破壊された <code>text</code> データまたは <code>image</code> データを指すテーブル・ローのロケーションがあることを示す。この情報は、問題の解決に役立つ場合がある。
100012	wrong type error	ページのフォーマットが不正であることを示す。たとえば、インデックスや <code>text</code> カラムまたは <code>image</code> カラムにデータ・ページが見つかった場合などである。
100013	non-OAM error	このエラーは、 <code>wrong type error</code> の特殊な場合である。現在のリリースでは独立した条件としてレポートされない。
100014	reused page error	ページが複数のチェーンによって到達されて、これらのチェーンが異なる複数のオブジェクトに属することを示す。このエラーは、破壊されたページ・チェーン・リンクによる不正なページ共有を示す。どちらかのテーブル、または両方のテーブル内のデータへのアクセスに影響する可能性がある。

type_code	type_name	説明
100015	page loop error	あるオブジェクトのページ・チェーンをたどっているときに、あるページに2回到達したことを示す。これは、ページ・チェーンにループがあることを示す。このオブジェクト内のデータに対するアクセス中、ループがあると、セッションが無期限ハングする可能性がある。
100016	OAM ring error	あるページが割り付けられたが、オブジェクトのページ・チェーンで到達されないことを示す。通常、これにより、割り付けられた記憶領域が使用できなくなる。
100017	OAM ring error	オブジェクト・アロケーション・マップのページ・リング・リンクが破壊されていることを示す。このことは、このオブジェクトのデータに対するアクセスには影響しないが、そのデータの挿入、削除、および更新に影響する可能性がある。
100018	missing OAM error	dbcc checkstorage によって、オブジェクト・アロケーション・マップに記録されなかったオブジェクトの割り付けが検出されたことを示す。このエラーは、今後の記憶領域の割り付けに影響する可能性がある破壊を示すが、現在格納されているデータへのアクセスにはおそらく影響しない。
100019	extra OAM error	このオブジェクトの割り付けがオブジェクト・アロケーション・マップに記録されたが、アロケーション・ページで確認されなかったことを示す。このエラーは、今後の記憶領域の割り付けに影響する可能性がある破壊を示すが、現在格納されているデータへのアクセスにはおそらく影響しない。
100020	check aborted error	dbcc checkstorage がテーブルまたはインデックスの検査を停止したことを示す。複数のフォールト・レポートの生成を防止するために、このエラーがレポートされずに、単一チェーンに対する検査オペレーションが停止した可能性がある。1つのオブジェクトにいくつかのページ・チェーンが含まれる場合、1つのチェーンに対する検査オペレーションが失敗しても、フォールト・スレッショルドを超えないかぎり、他のチェーンに対する検査オペレーションは中断されない。
100021	chain end error	チェーンの終了が破壊されていることを示す。このことは、ソフト・フォールトとして、dbcc checkstorage オペレーション中に多数のページによってチェーンが拡張されたか、またはトランケートされたことだけを示す場合がある。
100022	chain start error	チェーンの開始が破壊されたか、または予期されるロケーションにないことを示す。永続的なフォールトである場合は、オブジェクトに格納されたデータへのアクセスに影響する可能性がある。
100023	used count error	オブジェクト・アロケーション・マップ・ページに記録された使用済みページ・カウントと、アロケーション・ページを調べることによって判断される使用済みページのカウントが一致しないことを示す。
100024	unused count error	オブジェクト・アロケーション・マップ・ページに記録された予約済みだが未使用のページのカウントと、アロケーション・ページを調べることによって判断される予約済みだが未使用のページのカウントが一致しないことを示す。
100025	row count error	オブジェクト・アロケーション・マップ・ページに記録されたロー・カウントと、dbcc checkstorage によって判断されるロー・カウントが一致しないことを示す。
100026	serialloc error	ログ割り付けに適用される逐次割り付け規則の違反を示す。

<i>type_code</i>	<i>type_name</i>	説明
100027	text root error	text または image インデックスのルート・ページのフォーマット違反を示す。この検査は、dbcc textualloc によって実行されるルート・ページ検査に似ている。
100028	page misplaced	システム・テーブルの検査で、このオブジェクトのページが予期される場所から検出されなかったことを示す。このことは通常、sp_placeobject が過去のどこかで使用されたことを示す。dbcc_counters テーブルで、誤配置ページは、デバイスおよびパーティションごとにレポートされるのではなく、すべてまとめてカウントされる。
100029	page header error	他のタイプ・コードによって記述された種類以外のページ・ヘッダの内部矛盾を示す。このエラーの重大度は、ページのタイプと検出された矛盾によって異なる。
100030	page format error	他のタイプ・コードによって記述された種類以外のページ本文の内部矛盾を示す。このエラーの重大度は、ページのタイプと検出された矛盾によって異なる。
100031	page not allocated	dbcc checkstorage が、ページ・チェーンをたどって割り付けられていないページに達したことを示す。この条件は、このオブジェクトに格納されたデータへのアクセスに影響する可能性がある。
100032	page linkage error	dbcc checkstorage が、チェーンの内部ページの次または前のどちらかのリンクにフォールトを検出したことを示す。永続的なフォールトである場合は、オブジェクトに格納されたデータへのアクセスに影響する可能性がある。
100033	non-contiguous free-space error	ページの不連続な空き領域の値が無効または不一致であることを示す。
100034	insert free space error	ページの連続した空き領域の値が無効または不一致であることを示す。
100035	spacebits mismatch	ページ満杯インジケータの矛盾を示す。
100036	deleted row count error	ページの削除されたロー数の値が無効または不一致であることを示す。
100037	forwarded rows error	転送されたローのインジケータとページ上の転送されたロー数の不一致を示す。
100038	page header type error	ページ・ヘッダ・フォーマット・インジケータが正しく設定されていないことを示す。
100039	incorrect extent oampage	エクステント OAM ページ参照が正しく設定されていない。
100040	OAM page format error	最初以外の OAM ページに、ゼロ以外の、最初の OAM ページ固有のデータが含まれている。

モニタリング・テーブル

この章では、モニタリング・テーブルをアルファベット順に説明します。属性の列には、Adaptive Server がそのカラムを管理する方法が示されています。属性値の説明は、以下のとおりです。

- 「カウンタ」は、このカラム内の値が許容最大値である 2^{31} を超えた場合に循環する、つまりゼロに戻り再び増加する場合があることを示しています。noclear オプションを指定せずに `sp_sysmon` を実行すると、モニタ・カウンタは Adaptive Server によってリセットされます。Adaptive Server のバージョン 15.0.1 以降では、noclear オプションがデフォルトで `sp_sysmon` パラメータとして含まれています。15.0.1 より前のバージョンでは、モニタ・カウンタがリセットされないように `noclear` を指定する必要があります。

モニタリング・テーブルを使用している Adaptive Server 上で `sp_sysmon` を実行する場合、モニタ・カウンタがリセットされると結果に不整合が生じる可能性があります。

- “Null” は、そのカラムの値が null になる場合があることを示しています。
- 「リセット」は、モニタ・カウンタをクリアするように `sp_sysmon` を実行した場合、そのカラムがリセットされることを示しています (『パフォーマンス&チューニング・シリーズ: sp_sysmon による Adaptive Server の監視』を参照)。

monCachedObject

説明 現在データ・キャッシュにページが格納されているすべてのテーブル、パーティション、およびインデックスについての統計情報を格納します。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム monCacheObject のカラムは次のとおりです。

名前	データ型	属性	説明
CacheID	int		キャッシュのユニークな識別子。
InstanceID			クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
DBID	int		データベースのユニークな識別子。
IndexID	int		インデックスのユニークな識別子。
PartitionID	int		パーティションのユニークな識別子。分割されていないオブジェクトでは、ObjectID の値と同じ値。
CachedKB	int		オブジェクトが占有しているキャッシュのサイズ (キロバイト単位)。
CacheName	varchar(30)	Null	キャッシュの名前。
ObjectID	int	Null	オブジェクトのユニークな識別子。オブジェクトの記述子がサーバのメタデータ・キャッシュから削除されている場合は Null。この場合、PartitionID の値について、指定されているデータベース内の syspartitions にクエリを実行するとオブジェクト識別子を判断できる。
DBName	varchar(30)	Null	データベースの名前 (オブジェクトの記述子がサーバのメタデータ・キャッシュから削除されている場合は NULL)。
OwnerUserID	int	Null	オブジェクト所有者のユニークな識別子。
OwnerName	varchar(30)	Null	オブジェクト所有者の名前 (オブジェクトの記述子がサーバのメタデータ・キャッシュから削除されている場合は null)。
ObjectName	varchar(30)	Null	オブジェクトの名前 (オブジェクトの記述子がサーバのメタデータ・キャッシュから削除されている場合は null)。
PartitionName	varchar(30)	Null	オブジェクト・パーティションの名前 (オブジェクトの記述子がサーバのメタデータ・キャッシュから削除されている場合は null)。
ObjectType	varchar(30)	Null	オブジェクト・タイプ (オブジェクトがオープンしていない場合は null)。
TotalSizeKB	int	カウンタ、null	パーティションのサイズ (キロバイト単位)。
ProcessesAccessing	int	カウンタ、null	データ・キャッシュ内のこのオブジェクトのページに現在アクセスしているプロセスの数。

monCachePool

説明

すべてのデータ・キャッシュに割り付けられたすべてのプールについての統計情報を格納します。

注意 このモニタリング・テーブルでデータを収集するには、`enable monitoring` 設定パラメータを有効にしてください。

カラム

monCachePool のカラムは次のとおりです。

名前	データ型	属性	説明
CacheID	int		キャッシュのユニークな識別子。
InstanceID	int		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
IOBufferSize	int		プールの I/O バッファのサイズ (バイト単位)。
AllocatedKB	int		プールに割り付けられているバイト数。
PhysicalReads	int	カウンタ	ディスクからプールに読み込まれたバッファ数。
Stalls	int	カウンタ	このデータ・キャッシュのウォッシュ・エリアに使用できるクリーン・バッファがないために I/O オペレーションが遅延した回数。
PagesTouched	int	カウンタ	プール内で現在使用中のページ数。
PagesRead	int	カウンタ	プールに読み込まれたページ数。
BuffersToMRU	int	カウンタ	プールの最後に使用された部分でフェッチおよび置換されたバッファ数。
BuffersToLRU	int	カウンタ	プールの最も長い間使用されていない部分でフェッチおよび置換されたバッファ数 (使い捨て)。
CacheName	varchar(30)	Null	キャッシュの名前。

monCachedProcedures

説明

プロシージャ・キャッシュに現在格納されているすべてのストアド・プロシージャ、トリガ、およびコンパイル済みプランについての統計情報を格納します。

注意 このモニタリング・テーブルでデータを収集するには、**per object statistics active** 設定パラメータを有効にしてください。

カラム

monCacheProcedures のカラムは次のとおりです。

名前	データ型	属性	説明
ObjectID	int		プロシージャのユニークな識別子。
InstanceID	int		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
OwnerUID	int		オブジェクトの所有者のユニークな識別子。
DBID	int		オブジェクトが存在するデータベースのユニークな識別子。
PlanID	int		プロシージャ・キャッシュにあるオブジェクトのクエリ・プランのユニークな識別子。
MemUsageKB	int		プロシージャが使用したメモリ量 (キロバイト単位)。
CompileDate	datetime		プロシージャがコンパイルされた日付。
ObjectName	varchar(30)	Null	プロシージャの名前。
ObjectType	varchar(32)	Null	プロシージャのタイプ (ストアド・プロシージャ、トリガなど)。
OwnerName	varchar(30)	Null	オブジェクト所有者の名前。
DBName	varchar(30)	Null	データベースの名前。
RequestCnt	int4		このプロシージャがキャッシュから要求された回数。
TempdbRemapCnt	int4		このプロシージャがテンポラリ・データベースの ID に対して再マップされた回数。
AvgTempdbRemapTime	int4		テンポラリ・データベースの ID を再マップするのに費やした平均時間 (ミリ秒単位)。

monCachedStatement

説明

ステートメント・キャッシュに関する詳細なモニタリング情報を格納します。これには、以前に文を実行したときに使用されたリソース、文の実行頻度、特定のプランに対して有効になっている設定、文の同時使用数などに関する情報が含まれます。これらの情報は、トラブルシューティングを行うときや、どの文をキャッシュに残すかを決定するときに役立ちます。

monCachedStatement のカラムには、「カウンタ」(そのカラムにカウンタ値がある場合) と「リセット」(sp_sysmon を使用してそのカラムをリセットできる場合) の 2 種類の属性が許容されています。

注意 このモニタリング・テーブルでデータを収集するには、enable monitoring 設定パラメータと enable statement cache monitoring 設定パラメータを有効にし、statement cache size パラメータを 0 より大きい値に設定してください。

カラム

monCacheStatement のカラムは次のとおりです。

名前	データ型	属性	説明
SSQLID	int		キャッシュされた各文を示すユニークな識別子。この値は、monCachedStatement のプライマリ・キーとして扱われ、関数で使用される。 show_cached_text は、キャッシュ内の個々の文を参照するために SSQLID を使用する。
HashKey	int		キャッシュされた文の SQL テキストのハッシュ値。ハッシュ・キーは、文のテキストに基づいて生成され、他のモニタリング・テーブル内を検索するための近似キーとして使用できる。
UserID	int		キャッシュされた文を開始したユーザのユーザ ID。
SUserID	int		キャッシュされた文を開始したユーザのサーバ ID。
DBID	int		キャッシュされた文が取得されたデータベースのデータベース ID。
DBName	varchar(30)	Null	ステートメント・キャッシュ元であるデータベースの名前。
CachedDate	datetime		文が最初にキャッシュされたときの日付と時刻を表すタイムスタンプ。
LastUsedDate	datetime		キャッシュされた文が最後に使用されたときの日付と時刻を表すタイムスタンプ。この情報を CachedDate とともに使用すると、この文がどの程度の頻度で使用されているか、また、この文をキャッシュすると有益かどうかを判断できる。
CurrentUsageCount	int	カウンタ	キャッシュされた文の同時ユーザの数。
StatementSize	int		キャッシュされた文のサイズ (バイト単位) 。
MaxUsageCount	int	カウンタ	キャッシュされた文のテキストが同時にアクセスされた最大回数。

名前	データ型	属性	説明
SessionSettings			これらのセッションレベルの設定は、キャッシュされたそれぞれの文に関連付けられている。
ParallelDegree			この文用に格納されているクエリが使用する並列度。
QuotedIdentifier			<code>set quoted_identifier</code> を使用してコンパイルしたプランが有効かどうかを示す。
TransactionIsolationLevel			文をコンパイルしたトランザクション独立性レベル。
TransactionMode			文に対して「連鎖トランザクション・モード」が有効かどうかを示す。
SAAuthorization			<code>sa_role</code> 権限を使用してプランをコンパイルしたかどうかを示す。
SystemCatalogUpdates			プランをコンパイルしたときに <code>allow catalog updates</code> が有効な状態であったかどうかを示す。
ExecutionMetrics			<p>キャッシュされたプランが使用されると、実行コストが収集される。これらのコストは、論理 I/O (LIO) と物理 I/O (PIO) の値、実行時間、および経過時間に基づいて測定される。</p> <p><code>ExecutionMetrics</code> がレポートする測定基準は、QP 測定基準によってレポートされる測定基準と同じであるが、このテーブル内でレポートされるデータに対して QP 測定基準を有効にする必要はない。</p> <p><code>monCachedStatement</code> は、Adaptive Server の <code>metrics capture</code> 設定とは関係なく、キャッシュされた文に対して測定基準を独自に取得する。</p>
MetricsCount			この文に対して測定基準が集計された回数。
MaxElapsedTime	int		この文に対する最大の経過実行時間。
MinElapsedTime	int		この文に対する最小の経過実行時間。
AvgElapsedTime	int		この文に対する平均経過実行時間。
MaxLIO	int		この文の 1 回の実行中に発生する論理 I/O の最大数。
MinLIO	int		この文の 1 回の実行中に発生する論理 I/O の最小数。
AvgLIO	int		この文の実行中に発生する論理 I/O の平均数。
MaxPIO	int		この文の実行中に発生する物理 I/O の最大数。
MinPIO	int		この文の実行中に発生する物理 I/O の最小数。
AvgPIO	int		この文の実行中に発生する物理 I/O の平均数。
NumRecompilesPlanFlushes	int	カウンタ	キャッシュ内でプランが見つからなかったために、キャッシュされた文が再コンパイルされた回数。
NumRecompilesSchemaChanges	int	カウンタ	<p>スキーマの変更に基づいて文が再コンパイルされた回数。 <code>update statistics</code> をテーブルに対して実行すると、最適なプランが変更される場合がある。この変更は、重要度の低いスキーマ変更として処理される。</p> <p>ある文を何度も再コンパイルしている場合は、その特定の文をキャッシュすることが効果的でないと考えられる。この場合は、ステートメント・キャッシュからその文を削除して、より安定した別の文のために領域を空ける必要がある。</p>

名前	データ型	属性	説明
MaxPlanSize	int		使用されているときのプランのサイズ (キロバイト単位)。
MinPlanSize	int		使用されていないときのプランのサイズ (キロバイト単位)。
LastRecompiledDate	datetime		スキーマの変更、またはステートメント・キャッシュ内で文が見つからなかったことにより、文が再コンパイルされた最後の日付。
UseCount	int		キャッシュ後の文へのアクセス回数。
HasAutoParams	boolean		パラメータ化されたりテラルが文で使用されている場合は “true”、使用されていない場合は “false”。

monCIPC

説明

クラスタ環境に固有のテーブル - 現在のインスタンス、またはすべてのインスタンスを基準にした、クラスタ内でのメッセージング全体を要約した数字を示します。

monCIPC テーブルでは、システム・ビューが `cluster` に設定されている場合、クラスタ内のインスタンスごとに1つのローが返され、それ以外の場合、クエリが実行されたインスタンスについて1つのローが返されます。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム

monCIPC のカラムは次のとおりです。

名前	データ型	属性	説明
InstanceID	tinyint		クラスタ内でのインスタンスの ID。
ReceiveCount	int		このインスタンスが受信したメッセージ数。
TransmitCount	int		このインスタンスが送信したメッセージ数。
Multicast	int		クラスタ内の他のすべてのインスタンスに対して送信されたメッセージ数。
Synchronous	int		同期送信されたメッセージ数。
ReceiveSoftError	int		このインスタンスで受信したりカバリ可能なエラーの数。
ReceiveHardError	int		このインスタンスで受信したりカバリ不可能なエラーの数。
TransmitsSoftError	int		このインスタンスでのリカバリ可能な転送エラーの数。
TransmitHardError	int		このインスタンスでのリカバリ不可能な転送エラーの数。
Retransmits	int		このインスタンスが実行した再転送の数。
Switches	int		プライマリ相互接続ネットワークとセカンダリ相互接続ネットワーク間での切り替え回数。
FailedSwitches	int		プライマリ相互接続ネットワークとセカンダリ相互接続ネットワーク間での切り替えに失敗した回数。
RegularBuffersInUse	int4		現在割り付けられている CIPC の通常のバッファ・プールからのバッファ数。
FreeRegularBuffers	int4		CIPC の通常のバッファ・プールで使用可能なバッファ数。
MaxRegularBuffersInUse	int4		サーバが起動してから割り付けられた CIPC の通常のバッファ・プールからのバッファの最大数。
LargeBuffersInUse	int4		現在割り付けられている CIPC の大容量バッファ・プールからのバッファ数。
FreeLargeBuffers	int4		CIPC の大容量バッファ・プールで使用可能なバッファ数。
MaxLargeBuffersInUse	int4		サーバが起動してから割り付けられた CIPC の大容量バッファ・プールからのバッファの最大数。

monCIPCEndpoints

説明

クラスタ環境に固有のテーブル - クラスタ・インスタンス内にある各サブシステムのトラフィック・データを含む詳細な一覧を示します。

インスタンス内の論理エンドポイントごとに1つのローが返されます。システム・ビューが **cluster** に設定されている場合、クラスタ内のノードごとに1つのロー・セットが返されます。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム

monCIPCEndpoints のカラムは次のとおりです。

名前	データ型	属性	説明
InstanceID	tinyint		クラスタ内でのインスタンスの ID。
ReceiveCount	int		クラスタ内のこの論理エンドポイントが受信したメッセージ数。
TransmitCount	int		インスタンス内のこの論理エンドポイントが送信したメッセージ数。
ReceiveBytes	int		インスタンス内のこの論理エンドポイントが受信したバイト数。
TransmitBytes	int		インスタンス内のこの論理エンドポイントが送信したバイト数。
ReceiveQ	int		この論理エンドポイントに現在キューイングされているメッセージ数。
MaxReceiveQ	int		この論理エンドポイントにキューイングされ、現在までに検出されたメッセージの最大数。
DoneQ	int		この論理エンドポイントで、処理済みで次のアクションを待機しているメッセージの現在の数。
MaxDoneQ	int		この論理エンドポイントで現在までに検出された、処理済みで次のアクションを待機しているメッセージの最大数。
MaxRecvQTime	real4		現在の論理エンドポイントのキュー内で1つのメッセージが待機する最大時間 (ミリ秒単位)。
AvgRecvQTime	real4		現在の論理扱われ、のキュー内で1つのメッセージが待機する平均時間 (ミリ秒単位)。
EndPoint	varchar		CIPC エンドポイントの名前。

monCIPCLinks

説明

クラスタ環境に固有のテーブル – クラスタ内のインスタンスの間のリンクのステータスをモニタします。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム

monCIPCLinks のカラムは次のとおりです。

名前	データ型	属性	説明
InstanceID	tinyint		クラスタ内でのインスタンスの ID。
LocalInterface	varchar30		リンクのローカル・ネットワーク・エンドポイントの名前。サーバ名として <i>hosts</i> ファイルに含まれている名前と同じ名前。
RemoteInterface	varchar30		リンクのリモート・エンドポイントの名前。サーバ名として <i>hosts</i> ファイルに含まれている名前と同じ名前。
PassiveState	varchar10		リンク上のトラフィックでリストされている最新の状態。
PassiveStateAge	int		PassiveState カラムが更新されてから経過した時間 (ミリ秒単位)。
ActiveState	varchar10		アクティブなモニタリングによって検出された、使用された最新の状態 (リンク上にトラフィックがない場合)。
ActiveStateAge	int		ActiveState カラムが更新されてから経過した時間 (ミリ秒単位)。

monCIPCMesh

説明

クラスタ環境に固有のテーブルーインスタンス単位で、接続のメッシュ、つまり現在のインスタンスから同じクラスタ内にある他のすべてのインスタンスへの接続を要約した数字を示します。

設定された最大数を上限として、クラスタ内の他の各ノードへの4つの接続ごとに1つのローが返されます。システム・ビューが **cluster** に設定されている場合、クラスタ内のアクティブなインスタンスごとに1つのロー・セットが返されます。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム

monCIPCMesh のカラムは次のとおりです。

名前	データ型	属性	説明
InstanceID	tinyint		クラスタ内でのインスタンスの ID。
FarInstanceID	tinyint		クラスタ内の最終インスタンスのインスタンス番号。
Received	int		このインスタンスが FarInstanceID インスタンスから受信したメッセージ数。
Dropped	int		FarInstanceID インスタンスから受信したメッセージのうち、リソース不足のために削除されたメッセージ数。
Transmitted	int		FarInstanceID インスタンスに転送されたメッセージ数。
Resent	int		FarInstanceID インスタンスに再送されたメッセージ数。
Retry	int		FarInstanceID インスタンスに再送されたパケット数。
ControlRx	int		InstanceID インスタンスが受信した制御メッセージ数。
ControlTx	int		このメッシュで InstanceID インスタンスが送信した制御メッセージ数。
SendQ	int		このメッシュで FarInstanceID への送信を待機している現在のメッセージ数。
MaxSendQ	int		InstanceID インスタンスが起動してから現在までのこのメッシュの送信キュー内のパケットの最大数。
SentQ	int		InstanceID インスタンスが FarInstanceID インスタンスに送信したパケットのうち、FarInstanceID インスタンスによってまだ認識されていないパケットの数。
MaxSentQ	int		送信されたが送信通知が処理されていないメッセージの最大数。
MaxSendQTime	real		このメッシュの送信キュー内のメッセージを処理するために費やした時間の最大時間。ミリ秒単位。
AvgSendQTime	real		このメッシュの送信キュー内のメッセージを処理するために費やした時間の平均時間。ミリ秒単位。

名前	データ型	属性	説明
Mesh	varchar		接続のチャンネル名。次のいずれか。 <ul style="list-style-type: none">• 帯域外• メッセージ• 大規模なメッセージ• ダイレクト・メモリ・アクセス (DMA)
MinRTT	int		メッセージで検出された最小往復遅延 (ユーザ・データグラム・プロトコル (UDP : user datagram protocol) トランスポートのみに適用)。
MaxRTT	int		メッセージで検出された最大往復遅延 (UDP トランスポートのみに適用)。
AverageRTT	int		メッセージで検出された平均往復遅延 (UDP トランスポートのみに適用)。

monCLMObjectActivity

説明

クラスタ環境に固有のテーブル — クラスタ・ロック情報を収集します。
monCLMObjectActivity の機能は次のとおりです。

- master データベースおよびユーザ・データベースでのみオブジェクトのアクティビティを追跡します。
 - 物理ロック・アクティビティをパーティション・レベルで追跡します。
- データベースのクラスタ・オブジェクト・ロックの Object-PartitionID は 0 です。

注意 このモニタリング・テーブルでデータを収集するために、パラメータは必要ありません。

カラム

monCLMObjectActivity のカラムは次のとおりです。

カラム名	タイプ	属性	説明
InstanceID	int1		インスタンス ID。
DBID	int4		データベース ID。
Object_PartitionID	int4		ロック要求を行っているオブジェクトの ID。
LockRequests	int4		クラスタ・ロック要求数。
LocalMaster	int4		現在のインスタンスがロック・マスタであることをロック要求が検出した回数。 クラスタ内に「ロック・マスタ」になるインスタンスが 1 つある。クラスタ・ロックが必要な場合、インスタンスはロック・マスタに通知してロックを要求する。
Waited	int4		リモート・インスタンスで付与されて競合を発生しているロック要求の数。
Granted	int4		リモート・インスタンスで付与されて競合を発生していないロック要求の数。
RWConflictWaited	int4		リモート・インスタンスでタスクに付与された読み込み/書き込み競合ロックが原因で待機していたロック要求の数。
AvgRWConflictWaitTime	flt4		RWConflictWaited で記述された待機の実行に費やした平均時間。
MaxRWConflictWaitTime	flt4		RWConflictWaited で記述された待機の実行に費やした最大時間。
WWConflictWaited	int4		リモート・インスタンスでタスクに付与された、書き込みどうしの競合ロックが原因で待機していたロック要求の数。
AvgWWConflictWaitTime	flt4		WWConflictWaited で記述された待機の実行に費やした平均時間。
MaxWWConflictWaitTime	flt4		WWConflictWaited で記述された待機の実行に費やした最大時間。
ClusterMsgWaits	int4		クラスタ・メッセージングによる待機の数。

カラム名	タイプ	属性	説明
AvgClusterMsgWaitTime	flt4		クラスタ・メッセージングによる平均待機時間。
MaxClusterMsgWaitTime	flt4		クラスタ・メッセージングによる最大待機時間。
DowngradeReqRecv	int4		クラスタ・ロックの所有者が受信したダウングレード・サービス要求の数。
DowngradeReqRecvWithNoBlocker	int4		クラスタ・ロックの所有者が受信したブロッキング・タスク所有権のないダウングレード・サービス要求の数。
ClusterDeadlock	int4		複数のインスタンスが同じクラスタ・ロックを同時に取得しようとしたために発生したデッドロックの数。
Locktype	varchar(20)		ロックのタイプ。

monClusterCacheManager

説明 クラスタ環境に固有のテーブル。各インスタンスで実行されているクラスタ・キャッシュ・マネージャ・デーモンに関する診断情報を格納します。
monClusterCacheManager は、クラスタ全体にわたる情報をインスタンス単位でレポートします。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム monClusterCacheManager のカラムは次のとおりです。

名前	データ型	属性	説明
InstanceID	tinyint		クラスタ内でのインスタンスの ID。
RequestsQueued	int		クラスタ・キャッシュ・マネージャ・デーモンにキューイングされた要求数。
RequestsRequeued	int		クラスタ・キャッシュ・マネージャ・デーモンに再キューイングされた要求数。
RequestsServiced	int		クラスタ・キャッシュ・マネージャ・デーモンが処理した要求数。
DiskWrites	int		クラスタ・キャッシュ・マネージャ・デーモンが開始したディスクへの書き込みの数。
SleepCount	int		クラスタ・キャッシュ・マネージャ・デーモンがスリープした回数。
DaemonName	varchar		クラスタ・キャッシュ・マネージャ・デーモンの名前。
TransfersInitiated	int		クラスタ・キャッシュ・マネージャ・デーモンが開始した転送の数。
Downgrades	int		クラスタ・キャッシュ・マネージャ・デーモンが実行したダウングレードの数。
Releases	int		クラスタ・キャッシュ・マネージャ・デーモンが実行したリリースの数。
AvgServiceTime	int		要求を処理するのに費やした平均時間 (ミリ秒単位)。
MaxQSize	int		インスタンスが起動してからクラスタ・キャッシュ・マネージャ・デーモンにキューイングされた要求の最大数。

monCMSFailover

説明

クラスタ環境に固有のテーブル - クラスタ・メンバシップ・サービス (CMS : cluster membership service) によるエラーの検出、新しいクラスタ・ビューの取得、ハートビートの再同期、エラー・イベントの通知、およびエラー・イベントの完了が行われる時刻を追跡します。インスタンスごとに1つのローがあります。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム

monCMSFailover のカラムは次のとおりです。

カラム名	タイプ	属性	説明
InstanceID	tinyint1		フェールオーバを実行しているインスタンス。
FailedInstanceID	varchar(96)		エラーが発生したインスタンス ID のカンマ区切りのリスト。
FailDetectTime	datetime(8)		クラスタのエラーが検出された時刻。
InitViewTime	datetime(8)		最初のクラスタ・ビューが取得された時刻。
FinalViewTime	datetime(8)		最後のクラスタ・ビューが取得された時刻。
ResynchHBTime	datetime(8)		クラスタ全体のハートビートが再同期された時刻。
NotifyFailTime	datetime(8)		エラー・イベントが通知された時刻。
EventdoneTime	datetime(8)		最後のエラー・イベントが終了した時刻。

monDataCache

説明

Adaptive Server データ・キャッシュに関する統計情報を格納します。

注意 このモニタリング・テーブルでデータを収集するには、enable monitoring 設定パラメータを有効にしてください。

カラム

monDataCache のカラムは次のとおりです。

名前	データ型	属性	説明
CacheID	int		キャッシュのユニークな識別子。
InstanceID	int		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
RelaxedReplacement	int		キャッシュでリラックス・キャッシュ置換方式を使用しているかどうかを示す。
BufferPools	int		キャッシュ内のバッファ・プール数。
CacheSearches	int	カウンタ、リセット	このキャッシュに対するキャッシュ検索。
PhysicalReads	int	カウンタ、リセット	ディスクからキャッシュに読み込まれたバッファ数。
LogicalReads	int	カウンタ、リセット	キャッシュから取得されたバッファ数。
PhysicalWrites	int	カウンタ、リセット	キャッシュからディスクに書き込まれたバッファ数。
Stalls	int	カウンタ、リセット	ウォッシュ・エリアに使用できるクリーン・バッファがないために I/O オペレーションが遅延した回数。
CachePartitions	smallint		キャッシュ用として現在設定されているパーティション数。
CacheName	varchar(30)	Null	キャッシュの名前。

monDBRecovery

説明

クラスタ環境に固有のテーブル — クラスタ内のすべてのインスタンスからのローと、リカバリに必要な各データベースのローが含まれます。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム

monDBRecovery のカラムは次のとおりです。

カラム名	タイプ	属性	説明
DBID	int4		データベースのユニークな識別子。
InstanceID	int1		リカバリを実行したインスタンス (Cluster Edition のみに適用)。
MaxOpenXacts	int4		リカバリ中に参照されたオープン・トランザクションの最大数。
MaxPFTSEntries	int4		リカバリ中に参照された PFTS エントリの最大数。
Buckets	int4		バケットの数。
LogBTotPages	int4		ログ境界決定パスの間の log scan getpage 要求の数。
LogBTotAPFWaited	int4		ログ境界決定パスの間に進行中の I/O を検出した log scan getpage 要求の数。
LogBTotIO	int4		ログ境界決定パスの間の物理 I/O のある log scan getpage 要求の数。
AniTotRec	int4		リカバリ・プロセスによってスキャンされるログ・レコードの総数。
AniPhase1Recs	int4		フェーズ 1 リカバリ・プロセス内のログ・レコード数。
AniPhase1RedoRecs	int4		フェーズ 1 リカバリで再実行するログ・レコード数。
AniPhase2Recs	int4		フェーズ 2 リカバリ・プロセス内のログ・レコード数。
AniPhase2RedoRecs	int4		フェーズ 2 リカバリで再実行するログ・レコード数。
AniTotPages	int4		分析プロセス中の log scan getpage 要求の数。
AniTotAPFWaited	int4		分析パスの間に進行中の I/O を検出した log scan getpage 要求の数。
AniTotIO	int4		分析パスの間の物理 I/O のある log scan getpage 要求の数。
RedoOps	int4		再実行が検討されたオペレーションの総数。
RedoOpsNotRedonePFTS	int4		再実行が必要なかったオペレーション (PFTS チェック)。
RedoOpsRedonePFTS	int4		再実行が必要な可能性があったオペレーション (PFTS チェック)。
RedoOpsRedoneTS	int4		再実行が必要だったオペレーション (タイムスタンプ・チェック)。
RedoOpsNotRedoneTS	int4		再実行が必要でなかったオペレーション (タイムスタンプ・チェック)。
RedoLogTotPages	int4		再実行パスの間の log scan getpage 要求の数。
RedoLogTotAPFWaited	int4		再実行パスの間に進行中の I/O を検出した log scan getpage 要求の数。

カラム名	タイプ	属性	説明
RedoLogTotIO	int4		再実行パスの間の物理 I/O のある log scan getpage 要求の数。
RedoRecTotPage	int4		再実行パスの間の recovery pages getpage 要求の数。
RedoRecTotAPFWaited	int4		再実行パスの間に進行中の I/O を検出した recovery pages getpage 要求の数。
RedoRecTotIO	int4		再実行パスの間の物理 I/O のある recovery pages getpage 要求の数。
UndoRecsUndone	int4		取り消されたログ・レコード数。
UndoLogTotPages	int4		取り消しパスの間の log scan getpage 要求の数。
UndoLogTotAPFWaited	int4		取り消しパスの間に進行中の I/O を検出した log scan getpage 要求の数。
UndoLogTotIO	int4		取り消しパスの間の物理 I/O のある log scan getpage 要求の数。
UndoRecTotPages	int4		取り消しパスの間の recovery pages getpage 要求の数。
UndoRecTotAPFWaited	int4		取り消しパスの間に進行中の I/O を検出した recovery pages getpage 要求の数。
UndoRedTotIO	int4		取り消しパスの間の物理 I/O のある recovery pages getpage 要求の数。
DBName	varchar(30)		データベースの名前。
FailedInstanceID	int1		失敗したインスタンスの ID (Cluster Edition のみに適用)。
Command	varchar(30)		リカバリを実行中のプロセスによって実行された load database コマンド、load transaction コマンド、online database コマンド、mount database コマンド、および起動またはフェールオーバー・コマンドのいずれか。
RecType	varchar(30)		リカバリのタイプ。server start、load database、load transaction、または node failover のいずれか。
LobBStartTime	datetime(8)		ログ境界決定パスの開始時刻。
LogBEndTime	datetime(8)		ログ境界決定パスの終了時刻。
AnlStartTime	datetime(8)		分析パスの開始時刻。
AnlEndTime	datetime(8)		分析パスの終了時刻。
RedoStartTime	datetime(8)		再実行パスの開始時刻。
RedoEndTime	datetime(8)		再実行パスの終了時刻。
UndoStartTime	datetime(8)		取り消しパスの開始時刻。
UndoEndTime	datetime(8)		取り消しパスの終了時刻。

monDBRecoveryLRTypes

説明

クラスタ環境に固有のテーブル — リカバリ中に参照されたログ・レコードを追跡します。少なくとも1つのログ・レコードがリカバリによって参照された、各ログ・レコード・タイプのローが含まれます。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム

monDBRecoveryLRTypes のカラムは次のとおりです。

カラム名	タイプ	属性	説明
DBID	int4		データベースのユニークな識別子。
InstanceID	int1		リカバリを実行するインスタンス (Cluster Edition のみに適用)。
NumRecs	int4		リカバリ中に参照された、タイプ別のレコード数。
LogRecType	varchar(30)		ログ・レコード・タイプ。

monDBRecoveryLRTypes には、パラメータは必要ありません。

monDeadLock

説明

デッドロックに関する情報を提供します。deadlock pipe max messages を使用すると、返されるメッセージの最大数を調整できます。

monDeadLock は、履歴モニタリング・テーブルです。『パフォーマンス & チューニング・シリーズ：モニタリング・テーブル』を参照してください。

sp_monitor 'deadlock' を使用すると、現在のデッドロック・オプションを確認できます。deadlock パラメータを使用すると、サーバ・デッドロックの履歴を分析するために役立つ、monDeadLock に基づいたさまざまなレポートが提供されます。

注意 このモニタリング・テーブルでデータを収集するには、enable monitoring、deadlock pipe active、および deadlock max pipe messages の設定パラメータを有効にしてください。

カラム

monDeadLock のカラムは次のとおりです。

名前	データ型	属性	説明
DeadLockID	int		デッドロックのユニークな識別子。
VictimKPID	int		デッドロックのビクティム・プロセスのカーネル・プロセス ID (kpid)。
InstanceID	int		共有ディスク・クラスタ内のインスタンスの ID。
ResolveTime	datetime		デッドロックが解決された時刻。
ObjectDBID	int		オブジェクトが存在するデータベースのユニークなデータベース識別子。
PageNumber	int		ロックの要求されたページ番号 (該当する場合)。
RowNumber	int		ロックの要求されたロー番号 (該当する場合)。
HeldFamilyId	smallint		ロックを保持している親プロセスの spid。
HeldSPID	smallint		ロックを保持しているプロセスの spid。
HeldKPID	int		ロックを保持しているプロセスの kpid。
HeldProcDBID	int		ロックの保持を引き起こしたストアド・プロシージャが存在するデータベースのユニークな識別子 (該当する場合)。
HeldProcedureID	int		ロックの保持を引き起こしたストアド・プロシージャのユニークなオブジェクト識別子 (該当する場合)。
HeldBatchID	int		デッドロックの発生時にロックを保持していたプロセスによって実行されている SQL バッチの識別子。
HeldContextID	int		ロックを保持しているプロセスが (ロックの取得時ではなく) 別のプロセスにブロックされたときのユニークなコンテキスト識別子。
HeldLineNumber	int		ロックを保持しているプロセスが (ロックの取得時ではなく) 別のプロセスにブロックされたときに実行している文のバッチ内での行番号。
WaitFamilyId	smallint		ロックを待機している親プロセスの spid。
WaitSPID	smallint		ロックを待機しているプロセスの spid。

名前	データ型	属性	説明
WaitKPID	int		ロックを待機しているプロセスの kpid。
WaitTime	int		デッドロックが解決するまで待機中のプロセスがブロックされていた時間 (ミリ秒単位)。
ObjectName	varchar(30)	Null	オブジェクトの名前。
HeldUserName	varchar(30)	Null	ロックを保持しているユーザの名前。
HeldAppName	varchar(30)	Null	ロックを保持しているアプリケーションの名前。
HeldTranName	varchar(255)	Null	ロックが取得されたトランザクションの名前。
HeldLockType	varchar(20)	Null	保持されているロックのタイプ。
HeldCommand	varchar(30)		プロセスのカテゴリ、またはプロセスがブロックされたときに実行していたコマンド。
WaitUserName	varchar(30)	Null	ロックを要求したユーザの名前。
WaitLockType	varchar(20)	Null	要求されたロックのタイプ。
HeldSourceCodeID	varchar(30)		内部使用のみ。
WaitSourceCodeID	varchar(30)		内部使用のみ。

monDeviceIO

説明 データベース・デバイス上のアクティビティに関連する統計情報を返します。

注意 このモニタリング・テーブルでデータを収集するには、`enable monitoring` 設定パラメータを有効にしてください。

カラム `monDeviceIO` のカラムは次のとおりです。

名前	データ型	属性	説明
InstanceID	int		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
Reads	int	カウンタ、リセット	デバイスからの読み込み回数。
APFReads	int	カウンタ、リセット	デバイスからの非同期プリフェッチ (APF: asynchronous prefetch) 読み込み回数。
Writes	int	カウンタ、リセット	デバイスへの書き込み回数。
DevSemaphoreRequests	int	カウンタ、リセット	ミラーリングされたデバイスへの I/O 要求数 (ミラーリングされている場合)。
DevSemaphoreWaits	int	カウンタ、リセット	ミラーリングされたデバイスへの I/O 要求の同期を待機するよう強制されたタスク数 (ミラーリングされている場合)。
IOTime	int	カウンタ	I/O 要求の受け入れを待機した合計時間 (ミリ秒単位)。
LogicalName	varchar(30)	Null	デバイスの論理名。
PhysicalName	varchar(128)	Null	デバイスのフル・パス名。

monEngine

説明

Adaptive Server エンジンについての統計情報を提供します。

注意 このモニタリング・テーブルでデータを収集するには、`enable monitoring` 設定パラメータを有効にしてください。

カラム

monEngine のカラムは次のとおりです。

名前	データ型	属性	説明
EngineNumber	smallint		エンジンの番号。
InstanceID	int		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
CurrentKPID	smallint		現在実行中のプロセスのカーネル・プロセス識別子 (kpid)。
PreviousKPID	int		以前実行中であったプロセスの kpid。
CPUTime	int	カウンタ、リセット	エンジンの総実行時間 (秒単位)。
SystemCPUTime	int	カウンタ、リセット	エンジンがシステム・データベース・サービスを実行した時間 (秒単位)。
UserCPUTime	int	カウンタ、リセット	エンジンがユーザ・コマンドを実行した時間 (秒単位)。
IOCPUTime	int4		エンジンが発行済みの IO の完了を待機した時間 (秒単位)。
IdleCPUTime	int	カウンタ、リセット	エンジンがアイドル・スピン・モードだった時間 (秒単位)。
Yields	int	カウンタ、リセット	エンジンがオペレーティング・システムに CPU を解放した回数。アイドル期間中の CPU 解放率は、 <code>runnable process search count</code> を使用して変更する。
Connections	int	カウンタ	このエンジンが処理する接続数。
DiskIOChecks	int	カウンタ、リセット	エンジンが非同期ディスク I/O をチェックした回数。これらのチェックの頻度は、 <code>i/o polling process count</code> を使用して変更する。
DiskIOPolled	int	カウンタ、リセット	ポーリングは、ディスク I/O チェックが、通知されたが完了していない非同期 I/O を示した場合に発生する。
DiskIOCompleted	int	カウンタ、リセット	エンジンが未処理の非同期ディスク I/O についてポーリングしたとき、完了していた非同期ディスク I/O の数。
MaxOutstandingIOs	int		このエンジンによって開始され、完了していない I/O 要求の現在の数。
ProcessesAffinitied	int		このエンジンに関連付けられているプロセスの数。
ContextSwitches	int	カウンタ、リセット	コンテキストの切り替え回数。
HkgcMaxQSize	int		このエンジンで Adaptive Server がハウスキーピング・ガーベジ・コレクションにキューイングできる項目の最大数。
HkgcPendingItems	int		このエンジンのハウスキーピング・ガーベジ・コレクションでこれから収集される項目数。

名前	データ型	属性	説明
HkgcHWMItems	int		サーバの起動後、任意の時点でハウスキーピング・ガーベジ・コレクションにキューイングされた保留中の項目の最大数。
HkgcOverflows	int		キューのオーバーフローが原因でハウスキーピング・ガーベジ・コレクションにキューイングされなかった項目数。
Status	varchar(20)	Null	エンジンのステータス (オンライン、オフラインなど)。
Starttime	datetime	Null	エンジンがオンラインになった日付。
StopTime	datetime		エンジンがオフラインになった日付。
AffinitiedToCPU	int	Null	エンジンが結び付けられている CPU の数。
OSPID	int	Null	エンジンを実行中のオペレーティング・システム・プロセスの識別子。

monErrorLog

説明

Adaptive Server エラー・ログの最新のエラー・メッセージを返します。errorlog pipe max messages を使用すると、返されるメッセージの最大数を調整できます。『パフォーマンス&チューニング・シリーズ：モニタリング・テーブル』を参照してください。

注意 このモニタリング・テーブルでデータを収集するには、enable monitoring、errorlog pipe active、および errorlog pipe max messages の各設定パラメータを有効にしてください。

カラム

monErrorLog のカラムは次のとおりです。

名前	データ型	属性	説明
SPID	smallint		セッション・プロセス識別子 (spid)。
InstanceID	int		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
KPID	int		カーネル・プロセス識別子 (kpid)。
FamilyID	smallint		親プロセスの spid。
EngineNumber	smallint		プロセスが実行されたエンジン。
ErrorNumber	int		エラー・メッセージ番号。
Severity	int		エラーの重大度。
State	int		エラーのステータス。
Time	datetime		エラー発生時のタイムスタンプ。
ErrorMessage	varchar(512)	Null	エラー・メッセージのテキスト。

monFailoverRecovery

説明

クラスタ環境に固有のテーブル - クラスタ・ロック・マネージャ (CLM: cluster lock manager)、データベース・リカバリ、およびクラスタ・メンバシップ・サービス (CMS: cluster membership service) の各モジュールについて集計されたフェールオーバー・リカバリ診断情報が含まれます。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム

monFailoverRecovery のカラムは次のとおりです。

カラム名	タイプ	属性	説明
InstanceID	tinyint1		リカバリを実行しているインスタンス。
ModuleName	varchar(30)		モジュール名。CML、CMS、または Database のいずれか。
FailedInstanceID	tinyint1		失敗したインスタンスの ID。
StartTime	datetime(8)		モジュールのリカバリの開始時刻。
EndTime	datetime(8)		モジュールのリカバリの終了時刻。

monIOQueue

説明

デバイス I/O の統計を、各デバイス上の通常のデータベースとテンポラリ・データベースのデータ I/O とログ I/O として表示します。

注意 このモニタリング・テーブルでデータを収集するには、**enable monitoring** 設定パラメータを有効にしてください。

カラム

monIOQueue のカラムは次のとおりです。

名前	データ型	属性	説明
InstanceID	int		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
IOs	int	カウンタ	I/O オペレーションの総数。
IOTime	int	カウンタ	I/O 要求の受け入れを待機した時間 (ミリ秒単位)。
LogicalName	varchar(30)	Null	デバイスの論理名。
IOType	varchar(12)	Null	I/O のグループ化のカテゴリ。UserData、UserLog、TempdbData、TempdbLog のいずれか。

monLicense

説明

Adaptive Server によって現在チェックアウトされているすべてのライセンスのリストを示します。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム

monLicense のカラムは次のとおりです。

名前	データ型	属性	説明
InstanceID	int		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
Quantity	int		この機能に使用されるライセンス数。
Name	varchar(30)	Null	機能ライセンスの名前。
Edition	varchar(30)	Null	この機能がライセンスされている Adaptive Server のエディション。
Type	varchar(64)	Null	ライセンスの種類。
Version	varchar(16)	Null	使用している機能ライセンスのバージョン。
Status	varchar(30)	Null	この機能ライセンスのステータス (ライセンスが猶予期間中かどうか)。
LicenseExpiry	datetime	Null	ライセンスの有効期限の日付 (期限付きのライセンスの場合)。
GraceExpiry	datetime	Null	このライセンスの有効期限の日付 (ライセンスが猶予期間中の場合)。このライセンスが猶予期間中かどうかは、Status カラムを参照。
LicenseID	varchar(150)	Null	ライセンスの識別子。猶予期間中のライセンスには、この識別子がない場合もある。
Filter	varchar(14)	Null	この機能ライセンスの選択時に使用されるフィルタ。フィルタを変更するには、sp_lmconfig を使用する。
Attributes	varchar(64)	Null	ライセンス属性。これらの属性は「名前 = 値」のペアであり、指定されている場合、Adaptive Server の特定の特性を制限する。制限は次のとおり。 <ul style="list-style-type: none"> • ME = エンジンの最大数 • MC = 接続の最大数 • MS = ディスク領域の最大数 • MM = メモリの最大数 • CP = CPU の最大数

注意 monLicense は、mon_role パーミッションが不要で、任意のユーザが使用できます。

monLocks

説明

許可されたロックと保留中のロック要求のリストを返します。

注意 このモニタリング・テーブルでデータを収集するには、`enable monitoring` 設定パラメータを有効にしてください。

カラム

monLocks のカラムは次のとおりです。

名前	データ型	属性	説明
SPID	smallint		ロックを保持または要求しているプロセスのセッション・プロセス識別子。
InstanceID	int		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
KPID	int		カーネル・プロセス識別子。
DBID	int		このデータベース・オブジェクトのユニークな識別子。
ParentSPID	smallint		親プロセス ID。
LockID	int		ロック・オブジェクト ID。
Context	int		ロック・コンテキスト (ビット・フィールド)。これらの値は <code>syslocks</code> の <code>context</code> カラムの値と同じ。 <code>syslocks</code> の詳細については、『リファレンス・マニュアル：テーブル』を参照。
DBName	varchar(30)		ロックされているオブジェクトのデータベースの名前。 <code>monLocks</code> にクエリを実行したときデータベースがオープンでない場合、このカラムは NULL。
ObjectID	int	Null	オブジェクトのユニークな識別子。
LockState	varchar(20)	Null	ロックが許可されたかどうかを示す。値は次のとおり。 <ul style="list-style-type: none"> Granted Requested
LockType	varchar(20)	Null	ロックのタイプ。値は次のとおり。 <ul style="list-style-type: none"> Exclusive Shared Update
LockLevel	varchar(30)	Null	ロックが要求されたオブジェクトのタイプ。値は次のとおり。 <ul style="list-style-type: none"> Row Page Table Address
WaitTime	int	Null	ロック要求が許可されなかった時間 (秒数)。
PageNumber	int	Null	<code>LockLevel = 'PAGE'</code> のとき、ロックされるページ。
RowNumber	int	Null	<code>LockLevel = 'ROW'</code> のとき、ロックされるロー。

名前	データ型	属性	説明
BlockedBy	int		ロック要求がブロックされている場合、BlockedBy カラムは、このロック要求をブロックしているロックを保持しているプロセスのセッション・プロセス識別子。要求がブロックされていない場合は Null。
BlockedState	varchar(64)		保持されているロックが他のロック要求をブロックしている場合、またはロック要求がブロックされている場合、ロックのステータス。値は次のとおり。 <ul style="list-style-type: none">• Blocked• Blocking• Demand• Detached• Null (ブロック状態ではない場合)
SourceCodeID	varchar(30)		内部使用のみ。

monLogicalCluster

説明 クラスタ環境に固有のテーブル — システムで現在設定されている論理クラスタに関する情報を表示します。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム monLogicalCluster のカラムは次のとおりです。

名前	データ型	属性	説明
LCID	int		論理クラスタ ID。
Attributes	int		論理クラスタ属性のビットマスク。
ActiveConnections	int		この論理クラスタを使用しているアクティブな接続の数。
BaseInstances	tinyint		この論理クラスタのベース・インスタンスとして設定されているインスタンスの数。
ActiveBaseInstances	tinyint		この論理クラスタが現在アクティブであるベース・インスタンスの数。
FailoverInstances	tinyint		この論理クラスタのフェールオーバー・インスタンスとして設定されているインスタンスの数。
ActiveFailoverInstances	tinyint		この論理クラスタが現在アクティブであるフェールオーバー・インスタンスの数。
Name	varchar(30)		論理クラスタ名。
State	varchar(20)		現在のステータス。次のいずれか。 <ul style="list-style-type: none"> • Online • Offline • Failed • Inactive • Time_wait
DownRoutingMode	varchar(20)		ダウン・ルーティング・モードの設定。次のいずれか。 <ul style="list-style-type: none"> • System • Open • Disconnect
FailoverMode	varchar(20)		フェールオーバー・モードの設定。instance または cluster。
StartupMode	varchar(20)		起動モードの設定。automatic または manual。
SystemView	varchar(20)		システム・ビューの設定。instance または cluster。
Roles	varchar(20)		この論理クラスタの特別な役割のカンマ区切りのリスト。“system” 論理クラスタには、常にシステム役割がある。“open” 論理クラスタには、「オープンな」役割がある。system 論理クラスタにオープンな役割もある場合、このカラムの値は system, open。特別な役割がない論理クラスタは null 値を返す。
LoadProfile	varchar(30)		この論理クラスタに関連付けられている負荷プロファイル。

名前	データ型	属性	説明
ActionnRelease	varchar(20)		<p>この論理クラスタの現在のアクションの解放モード。値は次のとおり。</p> <ul style="list-style-type: none">• Manual• Automatic <p>Manual は、ユーザがアクション解放コマンドを実行してこのクラスタに対するアクションを解放する必要があることを示す。</p>
Gather	varchar(30)		<p>この論理クラスタへのルート指定可能な接続を自動的に収集するように論理クラスタが設定されているかどうかを示す。値は次のとおり。</p> <ul style="list-style-type: none">• Manual• Automatic

monLogicalClusterAction

説明 クラスタ環境に固有のテーブル。これらのアクションが起動されてから解放されるまでのローカル・クラスタに対するすべての管理アクションを示します。

`monLogicalClusterAction` でデータを収集するために必要な設定パラメータはありません。

カラム `monLogicalClusterAction` のカラムは次のとおりです。

名前	データ型	属性	説明
Handle	int		このアクションをキャンセルするために使用されるユニークなハンドル。
State	varchar(20)		アクションのステータス。active、complete、または canceled のいずれか。
LCID	int		このアクションが適用される論理クラスタ ID。
LogicalClusterName	varchar(30)		この論理クラスタ (ジョインを減らすために正規化解除されたクラスタ) の論理クラスタ名。
Action	varchar(15)		実行中のアクション。実行中のコマンドとそのスコープの組み合わせ。offline instance、failover cluster など。
FromInstances	varchar(96)		このコマンドとアクションの from instances のカンマ区切りのリスト (オフラインにするインスタンス)。
ToInstances	varchar(96)		このコマンドとアクションの to instances のカンマ区切りのリスト (オンラインにするインスタンス)。
InstancesWaiting	int		オフラインへの切り替えを待機しているインスタンスの数 (time_wait ステータスにある FromInstances のカウント)。
WaitType	varchar(20)		このアクションの現在の待機ステータス。値は、wait、until、または nowait のいずれか。
StartTime	datetime		コマンドが発行された日付と時刻。
Deadline	datetime		コマンドが終了する必要がある日付と時刻 (wait または until オプションに指定された時間値に基づく)。
CompleteTime	datetime		コマンドおよびアクションが完了した日付と時刻 (InstancesWaiting がゼロで、アクションのステータスが active から complete に変わった日付と時刻)。アクションが完了していない場合は NULL を返す。
ConnectionsRemaining	int		このコマンドの結果として移動される残りの接続数。
NonMigConnections	int		マイグレーション・プロトコルをサポートしていないために終了される接続数。
NonHACConnections	int		高可用性フェールオーバー・プロトコルをサポートしていない接続数。コマンドの終了時、これらの接続は切断され、フェールオーバーできない。

monLogicalClusterInstance

説明 クラスタ環境に固有のテーブル。インスタンスと論理クラスタ間の多対多の関係に関する情報を表示します。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム monLogicalClusterInstance のカラムは次のとおりです。

名前	データ型	属性	説明
LCID	int		論理クラスタ ID。
LogicalClusterName	varchar(30)		論理クラスタ名。
InstanceID	tinyint		クラスタ内でのインスタンスの ID。
InstanceName	varchar(30)		インスタンス名。
Type	varchar(20)		インスタンスのタイプ。
FailoverGroup	tinyint		このインスタンスがメンバであるフェールオーバー・グループ (フェールオーバー・インスタンスのみ)。
State	varchar(20)		論理クラスタに関するこのインスタンスのステータス。
ActiveConnections	int		このインスタンス上にあるこの論理クラスタのアクティブな接続の数。
NonMigConnections	int		接続マイグレーション・プロトコルをサポートしていないアクティブな接続の数。
NonHAConnections	int		高可用性フェールオーバー・プロトコルをサポートしていないアクティブな接続の数。
LoadScore	real		その論理クラスタに関連付けられた負荷プロファイルを使用するこのインスタンスの負荷スコア。

monLogicalClusterRoute

説明

クラスタ環境に固有のテーブル - 設定されているルートに関する情報 (アプリケーション、ログイン、およびエイリアス・バインド) を表示します。このモニタリング・テーブルにクエリを実行するために `mon_role` の役割は必要ありません。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム

monLogicalClusterRoute のカラムは次のとおりです。

名前	データ型	属性	説明
LCID	int		論理クラスタ ID。
LogicalClusterName	varchar(30)		論理クラスタ名。
RouteType	varchar(20)		ルート・タイプ。値は、application、login、または alias のいずれか。
RouteKey	varchar(30)		このルートに関連付けられたアプリケーション、ログイン、またはエイリアスの名前。

monNetworkIO

説明

Adaptive Server とクライアント接続間でのすべての通信のネットワーク I/O 統計情報を返します。

注意 このモニタリング・テーブルでデータを収集するには、`enable monitoring` 設定パラメータを有効にしてください。

カラム

monNetworkIO のカラムは次のとおりです。

名前	データ型	属性	説明
InstanceID	int		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
PacketsSent	int	カウンタ、リセット	送信したパケット数。
PacketsReceived	int	カウンタ、リセット	受信したパケット数。
BytesSent	int	カウンタ、リセット	送信したバイト数。
BytesReceived	int	カウンタ、リセット	受信したバイト数。

monOpenDatabases

説明 サーバのメタデータ・キャッシュ内に現在あるデータベースについてステータスおよび統計情報を提供します。

number of open databases の値が小さすぎる場合、Adaptive Server がデータベース記述子をメタデータ・キャッシュからフラッシュすることがあります。この場合、Adaptive Server はデータベース統計情報を消失しますが、次にそのデータベース記述子がメタデータ・キャッシュにインストールされると統計は再度開始されます。

注意 このモニタリング・テーブルでデータを収集するには、enable monitoring 設定パラメータを有効にしてください。

コラム monOpenDatabases のコラムは次のとおりです。

名前	データ型	属性	説明
DBID	int		データベースのユニークな識別子。
InstanceID	int		クラスタ環境に固有のコラム - 共有ディスク・クラスタ内のインスタンスの ID。
BackupInProgress	int		データベースのバックアップを現在実行中かどうかを示す。
LastBackupFailed	int		データベースの前のバックアップが失敗したかどうかを示す。
TransactionLogFull	int		データベースのトランザクション・ログが満杯かどうかを示す。
AppendLogRequests	int	カウンタ	データベース・トランザクション・ログに追加するときのセマフォ要求の数。
AppendLogWaits	int	カウンタ	ログ追加セマフォが付与されるまでタスクが待機した回数。
DBName	varchar(30)	Null	データベースの名前。
BackupStartTime	datetime	Null	データベースの前のフル・バックアップが開始された日付。
SuspendedProcesses	int	Null	データベースのトランザクション・ログが満杯であるために現在中断されているプロセスの数。
QuiesceTag	varchar(30)	Null	データベースが静止状態にある場合、このデータベースに対して quiesce database コマンドで使用されるタグ。
LastCheckpointTime	datetime	Null	このデータベースに対して前回実行されたチェックポイントの日付と時刻。
LastTranLogDumpTime	datetime	Null	このデータベースで成功した最新のトランザクション・ログ・ダンプの日付と時刻。

monOpenObjectActivity

説明 すべてのオープンなテーブルおよびインデックスについての統計情報を提供します。

注意 このモニタリング・テーブルでデータを収集するには、**enable monitoring**、**object lockwait timing**、および **per object statistics active** の各設定パラメータを有効にしてください。

カラム monOpenObjectActivity のカラムは次のとおりです。

名前	データ型	属性	説明
DBID	int		データベースのユニークな識別子。
ObjectID	int		オブジェクトのユニークな識別子。
IndexID	int		インデックスのユニークな識別子。
InstanceID	int		クラスタ環境に固有のカラム - インスタンスのユニークな識別子。
DBName	varchar(30)	Null	オブジェクトが存在するデータベースの名前。
ObjectName	varchar(30)	Null	オブジェクトの名前。
LogicalReads	int	カウンタ、null	ディスクから読み込まずに、このオブジェクトのバッファをバッファ・キャッシュから取得した合計回数。
PhysicalReads	int	カウンタ、null	ディスクから読み込まれたバッファ数。
APFReads	int	カウンタ、null	ディスクから読み込まれた APF バッファ数。
PagesRead	int	カウンタ、null	読み込まれたページの総数。
PhysicalWrites	int	カウンタ、null	ディスクに書き込まれたバッファの総数。
PagesWritten	int	カウンタ、null	ディスクに書き込まれたページの総数。
RowsInserted	int	カウンタ、null	挿入されたローの数。
RowsDeleted	int	カウンタ、null	削除されたローの数。
RowsUpdated	int	カウンタ、null	更新数。
Operations	int	カウンタ、null	オブジェクトへのアクセス回数。
LockRequests	int	カウンタ、null	オブジェクトに対するロック要求の数。
LockWaits	int	カウンタ、null	タスクがオブジェクト・ロックを待機した回数。
OptSelectCount	int	カウンタ、null	オブティマイザがこのインデックスをクエリ・プランで使用するために選択した回数。
LastOptSelectDate	datetime	Null	コンパイル中にプランのインデックスが選択された最後の日付。
UsedCount	int	カウンタ、null	実行中にオブジェクトがプランで使用された回数。
LastUsedDate	datetime	Null	実行中にプランでインデックスが使用された最後の日付。
HkgcRequests			オブジェクトにキューイングされたイベントの総数。値が大きい場合、指定されたオブジェクトのガーベジが大量に生成されている。

名前	データ型	属性	説明
HkgcPending			オブジェクトの保留中のイベントの数。値が大きい場合、まだ収集されていないガーベジが大量に存在し、ハウスキーピングによってこれからクリーニングされる。Adaptive Server をリポートすると、ハウスキーピング・キュー内のすべてのエントリが消失し、Adaptive Server の再起動時にそれらのページのガーベジは収集されない。
HkgcOverflows			オーバフロー・オブジェクト・イベントの数。値が大きい場合、ハウスキーピング・キューは満杯である。ハウスキーピングがジョブをスケジュールできないため、生成されたガーベジはクリーニングされない。
PhysicalLocks	int		クラスタ環境に固有のカラム - オブジェクトごとに要求される物理ロックの数。
PhsyicalLocksRetained	int		クラスタ環境に固有のカラム - 保持される物理ロックの数。各オブジェクトのロック・ヒット率を示すために使用する。ヒット率がよい場合、このオブジェクトはバランスよく分割されている。
PhysicalLocksRetainWaited	int4		クラスタ環境に固有のカラム - ロックが保持されるまで待機している物理ロック要求の数。
PhysicalLocksDeadlocks	int		クラスタ環境に固有のカラム - 要求された物理ロックがデッドロックを返した回数。sp_sysmon の Cluster Physical Locks サブセクションは、このカウンタを使用して、各オブジェクトの物理ロック取得中のデッドロックをレポートする。
PhysicalLocksWaited	int		クラスタ環境に固有のカラム - インスタンスが物理ロック要求を待機した回数。
PhysicalLocksPageTransfer	int		クラスタ環境に固有のカラム - インスタンスが物理ロックを要求したときに発生したページ転送数。sp_sysmon の Cluster Physical Locks サブセクションは、このカウンタを使用して、このオブジェクトのノードとの結び付き率としてノード間転送および物理ロック取得をレポートする。
TransferReqWaited	int4		クラスタ環境に固有のカラム - ページ転送を受信するまで物理ロック要求が待機した回数。
AvgPhysicalLocksWaitTime	int4		クラスタ環境に固有のカラム - 物理ロックが許可されるまでにクライアントが費やした平均時間。
AvgTransferReqWaitTime	int4		クラスタ環境に固有のカラム - ページ転送を受信するまでに物理ロック要求が待機した平均時間。
TotalServiceRequests	int4		クラスタ環境に固有のカラム - インスタンスのクラスタ・キャッシュ・マネージャによって処理された物理ロック要求の数。
PhysicalLocksDowngraded	int4		クラスタ環境に固有のカラム - インスタンスのクラスタ・キャッシュ・マネージャによって処理された物理ロック・ダウングレード要求の数。

名前	データ型	属性	説明
PagesTransferred	int4		クラスタ環境に固有のカラム - クラスタ・キャッシュ・マネージャによってインスタンスで転送されたページ数。
ClusterPageWrites	int4		クラスタ環境に固有のカラム - インスタンスのクラスタ・キャッシュ・マネージャによってディスクに書き込まれたページ数。
AvgServiceTime	int4		クラスタ環境に固有のカラム - インスタンスのクラスタ・キャッシュ・マネージャが費やした平均サービス時間。
AvgTimeWaitedOnLocalUsers	int4		クラスタ環境に固有のカラム - このインスタンスでユーザがページを使用しているために、インスタンスのクラスタ・キャッシュ・マネージャが待機した平均サービス時間。
AvgTransferSendWaitTime	int4		クラスタ環境に固有のカラム - インスタンスのクラスタ・キャッシュ・マネージャがページ転送に費やした平均サービス時間。
AvgIOServiceTime	int4		クラスタ環境に固有のカラム - ページ転送のためにインスタンスのクラスタ・キャッシュ・マネージャが使用した平均サービス時間。
AvgDowngradeServiceTime	int4		クラスタ環境に固有のカラム - 物理ロックをダウングレードするためにクラスタ・キャッシュ・マネージャが使用する平均サービス時間。

注意 ストアド・プロシージャまたはトリガのプランは複数回使用できるため、OptSelectCount カラムの値は UsedCount の値より小さい場合があります。また、Adaptive Server が実行時にクエリ・プランの特定の部分を実行しない場合があるため、UsedCount の値は OptSelectCount の値より小さい場合があります。

monOpenPartitionActivity

説明 サーバ上の各オープン・パーティションの使用に関する情報を提供します。

注意 このモニタリング・テーブルでデータを収集するには、`enable monitoring` および `per object statistics active` の各設定パラメータを有効にしてください。

コラム monOpenPartitionActivity のコラムは次のとおりです。

名前	データ型	属性	説明
DBID	int		データベースのユニークな識別子。
ObjectID	int		オブジェクトのユニークな識別子。
IndexID	int		インデックスのユニークな識別子。
PartitionID	int		パーティションのユニークな識別子。
InstanceID	int		共有ディスク・クラスタ内のインスタンスの ID。
DBName	varchar(30)	Null	オブジェクトが存在するデータベースの名前。
ObjectName	varchar(30)	Null	オブジェクトの名前。
PartitionName	varchar(30)	Null	パーティションの名前。
LogicalReads	int	カウンタ、null	読み込まれたバッファの総数。
PhysicalReads	int	カウンタ、null	ディスクから読み込まれたバッファ数。
APFReads	int	カウンタ、null	読み込まれた非同期プリフェッチ (APF) バッファの数。
PagesRead	int	カウンタ、null	読み込まれたページの総数。
PhysicalWrites	int	カウンタ、null	ディスクに書き込まれたバッファの総数。
PagesWritten	int	カウンタ、null	ディスクに書き込まれたページの総数。
RowsInserted	int	カウンタ、null	挿入されたローの数。
RowsDeleted	int	カウンタ、null	削除されたローの数。
RowsUpdated	int	カウンタ、null	更新数。
OptSelectCount	int	カウンタ、null	コンパイル中にプランのオブジェクトが選択された回数。
LastOptSelectDate	datetime	Null	コンパイル中にプランのインデックスが選択された最後の日付。
UsedCount	int	カウンタ、null	実行中にオブジェクトがプランで使用された回数。
LastUsedDate	datetime	Null	実行中にプランでインデックスが使用された最後の日付。
HkgcRequests			パーティションにキューイングされたイベントの総数。値が大きい場合、指定されたパーティションのガーベジが大量に生成される。

名前	データ型	属性	説明
HkgcPending			パーティションの保留中のイベント数。値が大きい場合、まだ収集されていないガーベジが大量に存在し、ハウスキーピングによってこれからクリーニングされる。Adaptive Server をリポートすると、ハウスキーピング・キュー内のすべてのエントリが消失し、Adaptive Server の再起動時にこれらのページのガーベジは収集されない。
HkgcOverflows			オーバフロー・パーティション・イベントの数。値が大きい場合、ハウスキーピング・キューは満杯である。ハウスキーピングがジョブをスケジュールできないため、生成されたガーベジはクリーニングされない。
PhysicalLocks	int		クラスタ環境に固有のカラム - オブジェクトごとに要求される物理ロックの数。
PhsyicalLocksRetained	int		保持されている物理ロックの数。各オブジェクトのロック・ヒット率を示すために使用する。ヒット率がよい場合、このオブジェクトはバランスよく分割されている。
PhysicalLocksRetainWaited	int4		クラスタ環境に固有のカラム - ロックが保持されるまで待機している物理ロック要求の数。
PhysicalLocksDeadlocks	int		クラスタ環境に固有のカラム - 要求された物理ロックがデッドロックを返した回数。sp_sysmon の Cluster Physical Locks サブセクションは、このカウンタを使用して、各オブジェクトの物理ロック取得中のデッドロックをレポートする。
PhysicalLocksWaited	int		クラスタ環境に固有のカラム - インスタンスが物理ロック要求を待機した回数。
PhysicalLocksPageTransfer	int		クラスタ環境に固有のカラム - インスタンスが物理ロックを要求したときに発生したページ転送数。sp_sysmon の Cluster Physical Locks サブセクションは、このカウンタを使用して、このオブジェクトのノードとの結び付き率としてノード間転送および物理ロック取得をレポートする。
TransferReqWaited	int4		クラスタ環境に固有のカラム - ページ転送を受信するまで物理ロック要求が待機した回数。
AvgPhysicalLockWaitTime	int4		クラスタ環境に固有のカラム - 物理ロックが許可されるまでにクライアントが費やした平均時間。
AvgTransferReqWaitTime	int4		クラスタ環境に固有のカラム - ページ転送を受信するまでに物理ロック要求が待機した平均時間。
TotalServiceRequests	int4		クラスタ環境に固有のカラム - インスタンスのクラスタ・キャッシュ・マネージャによって処理された物理ロック要求の数。
PhysicalLocksDowngraded	int4		クラスタ環境に固有のカラム - インスタンスのクラスタ・キャッシュ・マネージャによって処理された物理ロック・ダウングレード要求の数。

名前	データ型	属性	説明
PagesTransferred	int4		クラスタ環境に固有のカラム - クラスタ・キャッシュ・マネージャによってインスタンスで転送されたページ数。
ClusterPageWrites	int4		クラスタ環境に固有のカラム - インスタンスのクラスタ・キャッシュ・マネージャによってディスクに書き込まれたページ数。
AvgServiceTime	int4		クラスタ環境に固有のカラム - インスタンスのクラスタ・キャッシュ・マネージャが費やした平均サービス時間。
AvgTimeWaitedOnLocalUsers	int4		クラスタ環境に固有のカラム - このインスタンスでユーザがページを使用しているために、インスタンスのクラスタ・キャッシュ・マネージャが待機した平均サービス時間。
AvgTransferSendWaitTime	int4		クラスタ環境に固有のカラム - インスタンスのクラスタ・キャッシュ・マネージャがページ転送に費やした平均サービス時間。
AvgIOServiceTime	int4		クラスタ環境に固有のカラム - ページ転送のためにインスタンスのクラスタ・キャッシュ・マネージャが使用した平均サービス時間。
AvgDowngradeServiceTime	int4		クラスタ環境に固有のカラム - 物理ロックをダウングレードするためにクラスタ・キャッシュ・マネージャが使用する平均サービス時間。

注意 ストアド・プロシージャまたはトリガのプランは複数回使用できるため、OptSelectCount カラムの値は UsedCount の値より小さい場合があります。また、Adaptive Server が実行時にクエリ・プランの特定の部分を実行しない場合があるため、UsedCount の値は OptSelectCount の値より小さい場合があります。

monPCIBridge

説明

Java PCI Bridge についての情報が含まれます。このテーブルは、Java 環境についての情報を提供します。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム

monPCIBridge のカラムは次のとおりです。

名前	データ型	説明
InstanceID	tinyint	クラスタ環境に固有のカラム – 共有ディスク・クラスタ内のインスタンスの ID。
Status	char(10)	PCI Bridge の現在のステータス。値は次のとおり。 <ul style="list-style-type: none"> ACTIVE DOWN
ConfiguredSlots	int	設定されているスロットの数。max pci slots 設定パラメータを使用して設定する。
ActiveSlots	int	現在アクティブなスロットの数。
ConfiguredPCIMemoryKB	int	pci memory 設定パラメータを使用して PCI Bridge 用に設定されるメモリ合計。
UsedPCIMemoryKB	int	PCI Bridge とそのコンポーネントが現在使用しているメモリ合計。

monPCIEngine

説明

PCI Bridge とそのプラグインのエンジン情報を表示します。このテーブルは、Java 環境についての情報を提供します。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム

monPCIEngine のカラムは次のとおりです。

名前	データ型	説明
InstanceID	tinyint	クラスタ環境に固有のカラム – 共有ディスク・クラスタ内のインスタンスの ID。
Engine	int	エンジン番号。

名前	データ型	説明
Status	char(10)	エンジンのプラグインのステータス。値は次のとおり。 <ul style="list-style-type: none"> ACTIVE INIT
PLBStatus	char(10)	PCI Launcher Boss のステータス。値は次のとおり。 <ul style="list-style-type: none"> ACTIVE DOWN
NumberOfActiveThreads	int	現在 PCI Launcher Boss によって制御されているアクティブなスレッドの数。
PLBRequests	int	PCI Launcher Boss に対するネイティブ・スレッドの関数の実行要求の数。
PLBwakeupRequests	int	PCI Launcher Boss がネイティブ・スレッドの作業実行のためのウェイクアップを受信した回数。

monPCISlots

説明

PCI Bridge 内の各スロットにバインドされているプラグインについての情報が含まれます。このテーブルは、Java 環境についての情報を提供します。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム

monPCISlots のカラムは次のとおりです。

名前	データ型	説明
InstanceID	tinyint	クラスタ環境に固有のカラム – 共有ディスク・クラスタ内のインスタンスの ID。
Slot	int	アクティブなスロットの数。値は 1 ~ 31。
Status	char(10)	スロットのステータス。値は次のとおり。 <ul style="list-style-type: none"> INIT IN USE STOPPED
Modulename	varchar(30)	現在のスロットにバインドされている論理モジュール名。
engine	int	スロットに関連付けられているエンジン。

monPCM

説明

クラスタ環境に固有のテーブル - クラスタ内のピア・コーディネーション・モジュール (PCM : peer coordination module) クライアント・アクティビティ (送受信されたフラグメント数など) を追跡します。また、このテーブルには、各 PCM クライアントのローが含まれます。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム

monPCM のカラムは次のとおりです。

カラム名	タイプ	属性	説明
InstanceID	int1		情報が収集されたインスタンスの ID。
Sent	int4		モジュールごとの送信メッセージ数。
Fragments_sent	int4		モジュールごとの送信フラグメント数。
Fragments_received	int4		モジュールごとの受信フラグメント数。
Received	int4		モジュールごとの受信メッセージ数。
Reply	int4		モジュールごとの受信応答数。
Unicast	int4		モジュールごとの送信ユニキャスト・メッセージ数。
Mulicat	int4		モジュールごとの送信マルチキャスト・メッセージ数。
Sync	int4		モジュールごとの送信同期メッセージ数。
Async	int4		モジュールごとの送信非同期メッセージ数。
MinBytes	int4		メッセージごとに転送される最小バイト数。
AvgBytes	int4		メッセージごとに転送される平均バイト数。
MaxBytes	int4		メッセージごとに転送される最大バイト数。
MinDialog	int4		ダイアログの最小長。
AvgDialog	int4		ダイアログの平均長。
MaxDialog	int4		ダイアログの最大長。
Dialog	int4		ダイアログの数。
MinTimeSyncApi	flt4		モジュールごとに PCM レイヤ内で同期モードの PCM API に費やした最小時間。
AvgTimeSyncApi	flt4		モジュールごとに PCM レイヤ内で同期モードの PCM API に費やした平均時間。
MaxTimeSyncApi	flt4		モジュールごとに PCM レイヤ内で同期モードの PCM API に費やした最大時間。
MinTimeAsyncApi	flt4		モジュールごとに PCM レイヤ内で非同期モードの PCM API に費やした最小時間。
AvgTimeAsyncApi	flt4		モジュールごとに PCM レイヤ内で非同期モードの PCM API に費やした平均時間。
MaxTimeAsyncApi	flt4		モジュールごとに PCM レイヤ内で非同期モードの PCM API に費やした最大時間。

カラム名	タイプ	属性	説明
MinTimeCIPCMsgAlloc	flt4		モジュールごとに PCM レイヤ内で cipcmsg 割り付けに費やした最小時間。
AvgTimeCIPCMsgAlloc	flt4		モジュールごとに PCM レイヤ内で cipcmsg 割り付けに費やした平均時間。
MaxTimeCIPCMsgAlloc	flt4		モジュールごとに PCM レイヤ内で cipcmsg 割り付けに費やした最大時間。
MinTimeCIPCRecvCB	flt4		モジュールごとに cipc_recvcb に費やした最小時間。
AvgTimeCIPCRecvCB	flt4		モジュールごとに cipc_recvcb に費やした平均時間。
MaxTimeCIPCRecvCB	flt4		モジュールごとに cipc_recvcb に費やした最大時間。
MinTimeCIPCUnicastsmg	flt4		モジュールごとにユニキャスト・メッセージを送信中に CIPC に費やした最小時間。
AvgTimeCIPCUnicastsmg	flt4		モジュールごとにユニキャスト・メッセージを送信中に CIPC に費やした平均時間。
MaxTimeCIPCUnicastsmg	flt4		モジュールごとにユニキャスト・メッセージを送信中に CIPC に費やした最大時間。
MinTimeCIPCMulticastsmg	flt4		モジュールごとにマルチキャスト・メッセージを送信中に CIPC に費やした最小時間。
AvgTimeCIPCMulticastsmg	flt4		モジュールごとにマルチキャスト・メッセージを送信中に CIPC に費やした平均時間。
MaxTimeCIPCMulticastsmg	flt4		モジュールごとにマルチキャスト・メッセージを送信中に CIPC に費やした最大時間。
MinTimeClientRecvCB	flt4		モジュールごとに PCM レイヤ内でクライアント受信コールバックに費やした最小時間。
AvgTimeClientRecvCB	flt4		モジュールごとに PCM レイヤ内でクライアント受信コールバックに費やした平均時間。
MaxTimeClientRecvCB	flt4		モジュールごとに PCM レイヤ内でクライアント受信コールバックに費やした最大時間。
ModuleName	int4		PCM クライアントの名前。

monProcedureCache

説明 Adaptive Server プロシージャ・キャッシュに関する統計情報を返します。

注意 このモニタリング・テーブルでデータを収集するには、enable monitoring 設定パラメータを有効にしてください。

カラム monProcedureCache のカラムは次のとおりです。

名前	データ型	属性	説明
Requests	int	カウンタ、リセット	要求されたストアド・プロシージャ数。
Loads	int	カウンタ、リセット	キャッシュにロードされたストアド・プロシージャ数。
Writes	int	カウンタ、リセット	プロシージャが正規化され、ツリーが sysprocedures に書き戻された回数。
Stalls	int	カウンタ、リセット	ストアド・プロシージャをキャッシュにインストールするとき、プロセスが空きプロシージャ・キャッシュ・バッファを待機した回数。
InstanceID	int		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。

monProcedureCacheMemoryUsage

説明

プロシージャ・キャッシュ・アロケータごとに1つのローが含まれます。アロケータはアロケータ ID で識別されます。このアロケータ ID は、Adaptive Server の内部 ID です。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム

monProcedureCacheMemoryUsage のカラムは次のとおりです。

名前	データ型	属性	説明
InstanceID	tinyint		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
AllocatorID	int		アロケータ ID。
ModuleID	int		モジュール ID (Adaptive Server の内部 ID)。
Active	int		このアロケータに現在割り付けられているメモリ・ページ (2KB) の数。
HWM	int		サーバが起動してから割り付けられたメモリ・ページの最大数。
ChunkHWM	int		サーバが起動してから割り付けられた連続するメモリ・ページの最大数。
AllocatorName	varchar(30)		アロケータの名前。
NumReuseCaused	int	Null	このアロケータで置換が行われた回数。

monProcedureCacheModuleUsage

説明

プロシージャ・キャッシュからメモリを割り付けるモジュールごとに1つのローが含まれます。モジュールは、モジュール ID で識別され、Adaptive Server プロシージャ・キャッシュ管理の内部の機能領域分類です。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム

monProcedureCacheModuleUsage のカラムは次のとおりです。

名前	データ型	属性	説明
InstanceID	tinyint		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
ModuleID	int		モジュール ID。
Active	int		このモジュールに現在割り付けられているメモリ・ページ (2KB) の数。
HWM	int		サーバが起動してから割り付けられたメモリ・ページの最大数。
NumPagesReused	int	Null	このモジュールに割り付けられているページ数。
ModuleName	varchar(30)		モジュール名。

monProcess

説明 現在実行中または待機中のプロセスについての詳細な統計情報を提供します。

注意 このモニタリング・テーブルでデータを収集するには、`enable monitoring` および `wait event timing` の各設定パラメータを有効にしてください。

カラム monProcess のカラムは次のとおりです。

名前	データ型	属性	説明
SPID	smallint		セッション・プロセス識別子。
InstanceID	int		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
KPID	int		カーネル・プロセス識別子。
ServerUserID	int		このプロセスに関連付けられているユーザのサーバ・ユーザ ID (SUID)。
BatchID	int		実行文を含む SQL バッチのユニークな識別子。
ContextID	int		実行中のクエリによって、ストアド・プロシージャ、トリガ、即時実行、遅延コンパイル、その他のコンパイル済みオブジェクトが実行されるたびに生成されるユニークな識別子。
LineNumber	int		SQL バッチ内の現在実行されている文の行番号。
SecondsConnected	int		この接続が確立してから経過した秒数。
DBID	int		プロセスが使用しているデータベースのユニークな識別子。
EngineNumber	smallint		プロセスが実行されているエンジンのユニークな識別子。
Priority	int		プロセスの実行優先度。
FamilyID	int	Null	親プロセスの spid (ワーカー・プロセスの場合)。
Login	varchar(30)	Null	ログイン・ユーザ名。
Application	varchar(30)	Null	アプリケーション名。アプリケーションのログイン構造内で名前が設定されない場合、ブランクになる場合もある。
Command	varchar(30)	Null	プロセスのカテゴリ、またはプロセスが現在実行中のコマンド。
NumChildren	int	Null	子プロセス数 (並列クエリを実行中の場合)。
SecondsWaiting	int	Null	プロセスが待機している秒数 (現在、別のプロセスが保持するロックによってプロセスがブロックされている場合)。
WaitEventID	int	Null	プロセスが待機しているイベントのユニークな識別子 (現在、プロセスが待機状態の場合)。
BlockingSPID	int	Null	このプロセスが要求したロックを保持しているプロセスのセッション・プロセス識別子 (ロックを待機中の場合)。
BlockingXLOID	int	Null	このプロセスが要求したロックのユニークなロック識別子 (ロックを待機中の場合)。
DBName	varchar(30)	Null	プロセスが現在使用しているデータベースの名前。
EngineGroupName	varchar(30)	Null	プロセスのエンジン・グループ。
ExecutionClass	varchar(30)	Null	プロセスの実行クラス。
MasterTransactionID	varchar(255)	Null	プロセスがオープンしたトランザクションの名前。

monProcessActivity

説明 プロセス・アクティビティについての詳細な統計情報を提供します。

注意 このモニタリング・テーブルでデータを収集するには、`enable monitoring`、`object lockwait timing`、および `wait event timing` の各設定パラメータを有効にしてください。

カラム `monProcessActivity` のカラムは次のとおりです。

名前	データ型	属性	説明
SPID	smallint		セッション・プロセス識別子。
InstanceID	int		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
KPID	int		カーネル・プロセス識別子。
ServerUserID	int		このプロセスを実行しているユーザのサーバ・ユーザ識別子 (SUID)。 <code>ServerUserID</code> の値は、 <code>syslogins.suid</code> カラムに一致する。対応する名前を取得するには <code>suser_name</code> 関数を使用する。
CPUTime	int	カウンタ	プロセスに使用された CPU 時間 (ミリ秒単位)。
WaitTime	int	カウンタ	プロセスが待機した時間 (ミリ秒単位)。
PhysicalReads	int	カウンタ	ディスクから読み込まれたバッファ数。
LogicalReads	int	カウンタ	キャッシュから読み込まれたバッファ数。
PagesRead	int	カウンタ	読み込まれたページ数。
PhysicalWrites	int	カウンタ	ディスクに書き込まれたバッファ数。
PagesWritten	int	カウンタ	書き込まれたページ数。
MemUsageKB	int		プロセスに割り付けられたメモリ量 (バイト単位)。
LocksHeld	int		プロセスが現在保持しているロック数。
TableAccesses	int	カウンタ	インデックスを使用しないで Adaptive Server が取得した、読み込まれたページ数。
IndexAccesses	int	カウンタ	インデックスを使用して Adaptive Server が取得した、読み込まれたページ数。
WorkTables	int	カウンタ	プロセスが作成したワーク・テーブルの総数。
TempDbObjects	int	カウンタ	プロセスが作成したテンポラリ・テーブルの総数。
ULCBytesWritten	int	カウンタ	プロセスのユーザ・ログ・キャッシュに書き込まれたバイト数。
ULCFlushes	int	カウンタ	ユーザ・ログ・キャッシュがフラッシュされた合計回数。
ULCFlushFull	int	カウンタ	満杯のためユーザ・ログ・キャッシュがフラッシュされた回数。
ULCMaxUsage	int		プロセスによるユーザ・ログ・キャッシュの最大使用量 (バイト単位)。
ULCCurrentUsage	int		プロセスによるユーザ・ログ・キャッシュの現在の使用量 (バイト単位)。
Transactions	int	カウンタ	プロセスが開始したトランザクション数。
Commits	int	カウンタ	プロセスがコミットしたトランザクション数。
Rollbacks	int	カウンタ	プロセスがロールバックしたトランザクション数。

monProcessLookup

説明 サーバ上の各プロセスに関する識別情報を提供します。各プロセスのアクティビティに関する統計情報については、「[monProcessActivity](#)」(171 ページ)を参照してください。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

コラム monProcessLookup のコラムは次のとおりです。

名前	データ型	属性	説明
SPID	smallint		セッション・プロセス識別子。
InstanceID	int		クラスタ環境に固有のコラム – 共有ディスク・クラスタ内のインスタンスの ID。
KPID	int		カーネル・プロセス識別子。
Login	varchar(30)	Null	ログイン・ユーザ名。
Application	varchar(30)	Null	アプリケーション名。
ClientHost	varchar(30)	Null	クライアントのホスト名。
ClientIP	varchar(24)	Null	クライアントの IP アドレス。
ClientOSPID	varchar(30)	Null	クライアント・アプリケーションのオペレーティング・システム・プロセス識別子。

monProcessMigration

説明 クラスタ環境に固有のテーブル – 現在マイグレートしている接続に関する情報を表示します。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

コラム monProcessMigration のコラムは次のとおりです。

名前	データ型	属性	説明
SPID	int4		保留中のマイグレーション・セッション・プロセス ID。
KPID	int4		カーネル・プロセス ID。
LogicalCluster	varchar(30)		現在の論理クラスタ。
Instance	varchar(30)		現在のインスタンス。
MigrationLogicalCluster	varchar(30)		マイグレーション論理クラスタ。
MigrationInstance	varchar(30)		マイグレーション・インスタンス。
Command	varchar(30)		マイグレーション・トリガ。

monProcessNetIO

説明 各プロセスのネットワーク I/O アクティビティ情報を提供します。

注意 このモニタリング・テーブルでデータを収集するには、`enable monitoring` 設定パラメータを有効にしてください。

カラム `monProcessNetIO` のカラムは次のとおりです。

名前	データ型	属性	説明
SPID	smallint		セッション・プロセス識別子。
InstanceID	int		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
KPID	int		カーネル・プロセス識別子。
NetworkPacketSize	int		セッションで現在使用しているネットワーク・パケット・サイズ。
PacketSent	int	カウンタ	送信したパケット数。
PacketsReceived	int	カウンタ	受信したパケット数。
BytesSent	int	カウンタ	送信したバイト数。
BytesRecieved	int	カウンタ	受信したバイト数。
NetworkEngineNumber	smallint		このプロセスがネットワーク・エンジンとして使用しているエンジンの数。

monProcessObject

説明

プロセスが現在アクセスしているオブジェクトについての統計情報を提供します。

注意 このモニタリング・テーブルでデータを収集するには、**enable monitoring** および **per object statistics active** の各設定パラメータを有効にしてください。

カラム

monProcessObject のカラムは次のとおりです。

名前	データ型	属性	説明
SPID	smallint		セッション・プロセス識別子。
InstanceID	int		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
KPID	int		カーネル・プロセス識別子。
DBID	int		オブジェクトが存在するデータベースのユニークな識別子。
ObjectID	int		オブジェクトのユニークな識別子。
PartitionID	int		パーティションのユニークな識別子。
IndexID	int		インデックスのユニークな識別子。
OwnerUserID	int		オブジェクト所有者のユーザ識別子。
LogicalReads	int	カウンタ	キャッシュから読み込まれたバッファ数。
PhysicalReads	int	カウンタ	ディスクから読み込まれたバッファ数。
PhysicalAPFReads	int	カウンタ	ディスクから読み込まれた非同期プリフェッチ・バッファ数。
DBName	varchar(30)	Null	データベースの名前。
ObjectName	varchar(30)	Null	オブジェクトの名前。
PartitionName	varchar(30)	Null	パーティションの名前。
ObjectType	varchar(30)	Null	オブジェクトのタイプ。
PartitionSize	int	カウンタ、 null	パーティションのサイズ (キロバイト単位)。

monProcessProcedures

説明 プロセスによって実行されているすべてのプロシージャのリストを返します。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム monProcessProcedures のカラムは次のとおりです。

名前	データ型	属性	説明
SPID	smallint		セッション・プロセス識別子。
InstanceID	int		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
KPID	int		カーネル・プロセス識別子。
DBID	int		オブジェクトのデータベースのユニークな識別子。
OwnerUID	int		オブジェクト所有者のユニークな識別子。
ObjectID	int		プロシージャのユニークな識別子。
PlanID	int		クエリ・プランのユニークな識別子。
MemUsageKB	int		プロシージャによって使用されるメモリ量 (キロバイト単位)。
CompileDate	datetime		プロシージャがコンパイルされた日付。
ContextID	int		実行中のクエリによって、ストアド・プロシージャ、トリガ、即時実行、遅延コンパイル、その他のコンパイル済みオブジェクトが実行されるたびに生成されるユニークな識別子。
LineNumber	int		現在実行中のプロシージャの行。
DBName	varchar(30)	Null	プロシージャを含むデータベースの名前。
OwnerName	varchar(30)	Null	オブジェクトの所有者の名前。
ObjectName	varchar(30)	Null	プロシージャの名前。
ObjectType	varchar(32)	Null	プロシージャのタイプ (ストアド・プロシージャ、トリガなど)。

monProcessSQLText

説明

プロセスが現在実行している SQL テキストを提供します。max SQL text monitored を使用すると、SQL テキストの最大サイズを調整できます。

monProcessSQLText は、プロセスが実行する SQL テキスト・バッチのローごとに 1 つのローを返します (SPID で指定)。つまり、バッチに 3 つのローがある場合、monProcessSQLText は、その結果セットで 3 つのローを返します。LineNumber の値は、バッチ内の行数を示します。1 つのローの長さが 255 バイトを超えた場合、monProcessSQLText は、複数のローを返し、LineNumber の値はすべてのローで同じになります。ただし、SequenceInLine の値は、各ローで異なります。

注意 このモニタリング・テーブルでデータを収集するには、enable monitoring、SQL batch capture、および max SQL text monitored の各設定パラメータを有効にしてください。

カラム

monProcessSQLText のカラムは次のとおりです。

名前	データ型	属性	説明
SPID	smallint		セッション・プロセス識別子。
InstanceID	int		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
KPID	int		カーネル・プロセス識別子。
ServerUserID	int		この SQL を実行しているユーザのサーバ・ユーザ識別子 (SUID)。ServerUserID の値は、syslogins.suid カラムの値に一致する。対応する名前を取得するには suser_name 関数を使用する。
BatchID	int		SQL テキストを含む SQL バッチのユニークな識別子。
LineNumber	int		ローの SQL テキストの SQL バッチ行番号。
SequenceInLine	int		各ローは、増加するユニークな SequenceInLine 値を持つ。SQL テキストの長さが 255 バイトを超える場合、テキストは複数のローに分割される。
SQLText	varchar(255)	Null	実行中のテキスト。

monProcessStatement

説明 現在実行中の文についての情報を提供します。

注意 このモニタリング・テーブルでデータを収集するには、`enable monitoring`、`statement statistics active`、および `wait event timing` の各設定パラメータを有効にしてください。

カラム `monProcessStatement` のカラムは次のとおりです。

名前	データ型	属性	説明
SPID	smallint		セッション・プロセス識別子。
InstanceID	int		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
KPID	int		カーネル・プロセス識別子。
DBID	int		プロセスが現在使用しているデータベースのユニークな識別子。
ProcedureID	int		ストアド・プロシージャのユニークな識別子。
PlanID	int		プロセスが実行中のプランのユニークな識別子。
BatchID	int		文が実行されているプロセスのバッチ番号。
ContextID	int		プロシージャのスタック・フレーム (プロシージャの場合)。
LineNumber	int		SQL バッチ内の文の行番号。
CPUTime	int	カウンタ	文に使用された CPU 時間 (ミリ秒単位)。
WaitTime	int	カウンタ	タスクが文の実行中に待機した時間 (ミリ秒単位)。
MemUsageKB	int		文の実行に使用したメモリ量 (キロバイト単位)。
PhysicalReads	int	カウンタ	ディスクから読み込まれたバッファ数。
LogicalReads	int	カウンタ	キャッシュから読み込まれたバッファ数。
PagesModified	int	カウンタ	文によって修正されたページ数。
PacketsSent	int	カウンタ	Adaptive Server が送信したネットワーク・パケット数。
PacketsReceived	int	カウンタ	Adaptive Server が受信したネットワーク・パケット数。
NetworkPacketSize	int		セッションで現在設定されているネットワーク・パケットのサイズ (バイト単位)。
PlansAltered	int	カウンタ	実行時に変更されたプラン数。
RowsAffected	int		現在の文の影響を受けたローの数。非効率なクエリ・プランを使用しているクエリでは、多くの場合、返されるローごとの論理 I/O の数値が高い。
DBName	varchar(30)		このプロセスを実行中のデータベースの名前。プロセスがストアド・プロシージャまたはその他のコンパイル済みオブジェクトを実行している場合、データベース名は、そのオブジェクトのデータベースの名前になる。
StartTime	datetime	Null	文の実行が開始された日付。

monProcessWaits

説明 サーバ上で現在のプロセスが待機しているすべての待機イベントのリストを提供します。Waits 値がゼロより大きい待機イベントのみを返します。

注意 このモニタリング・テーブルでデータを収集するには、**enable monitoring**、**process wait events**、および **wait event timing** の各設定パラメータを有効にしてください。

コラム monProcessWaits のコラムは次のとおりです。

名前	データ型	属性	説明
SPID	smallint		セッション・プロセス識別子。
InstanceID	int		クラスタ環境に固有のコラム - 共有ディスク・クラスタ内のインスタンスの ID。
KPID	int		カーネル・プロセス識別子。
ServerUserID	int		このプロセスに関連付けられているユーザのサーバ・ユーザ ID (SUID)。
WaitEventID	smallint		待機イベントのユニークな識別子。
Waits	int	カウンタ	プロセスがイベントを待機した回数。
WaitTime	int	カウンタ	プロセスがイベントを待機した時間 (ミリ秒単位)。

WaitEventInfo には、各待機イベントの説明が含まれています。各モニタリング・テーブルの WaitEventID コラムをジョインすると、このデータを表示できます。

待機イベントの選択の説明については、『パフォーマンス&チューニング・シリーズ：モニタリング・テーブル』を参照してください。

monProcessWorkerThread

説明 現在設定されている各ワーカー・プロセスのアクティビティの統計情報を提供します。

注意 このモニタリング・テーブルでデータを収集するには、`enable monitoring` 設定パラメータを有効にしてください。

カラム monProcessWorkerThread のカラムは次のとおりです。

名前	データ型	属性	説明
SPID	smallint		セッション・プロセス識別子。
InstanceID	int		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
KPID	int		カーネル・プロセス識別子。
ThreadsActive	int		プロセスによって現在使用されているワーカー・スレッドの数。
MaxParallelDegree	smallint		セッションの <code>set parallel_degree</code> オプションを使用して設定される、このタスクで使用できる最大並列度。または、 <code>max parallel degree</code> の現在の Run Value。
MaxScanParallelDegree	smallint		セッションの <code>set scan_parallel_degree</code> オプションを使用して設定される、このタスクで使用できるスキャンの最大並列度。設定されない場合は、 <code>max scan parallel degree</code> の現在の Run Value。
ParallelQueries	int	カウンタ	このプロセスによって実行された並列クエリの総数。
PlansAltered	int	カウンタ	このプロセス用に「最適な」プランから変更されたプランの数。Adaptive Server で、最適な並列度でクエリを実行するために使用できるワーカー・スレッドの数が不足している場合、プランは変更される。
FamilyID	int	Null	親プロセスの spid (ワーカー・プロセスの場合)。

monState

説明

Adaptive Server 全体のステータスについての情報を提供します。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム

monState のカラムは次のとおりです。

名前	データ型	属性	説明
InstanceID	int		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
LockWaitThreshold	int		ブロックされたプロセスとしてカウントされ、LockWaits カラムにレポートされるまでプロセスがロックを待機する時間 (秒単位)。LockWaitThreshold のデフォルト値は 5 秒である。クエリの where 句 (LockWaitThreshold=30 など) に値が指定されていない場合、デフォルト値が使用される。
LockWaits	int		LockWaitThreshold の値よりも長い時間ロックを待機したプロセスの数。
DaysRunning	int		Adaptive Server が稼働している日数。
CheckPoints	int		現在実行中のチェックポイントがあるかどうかを示す。
NumDeadlocks	int	カウンタ	発生したデッドロックの総数。
Diagnostic Dumps	int		このサーバの共有メモリ・ダンプが現在実行中かどうかを示す。
Connections	int		アクティブなインバウンド接続数。
MaxRecovery	int		システム障害が発生した場合に Adaptive Server がリカバリ処理を完了するために使用するデータベースあたりの最大時間 (分単位)、および recovery interval in minutes 設定オプションの現在の Run Value。
Transactions	int4		トランザクション実行数 (サーバワイド)。
StartDate	datetime		Adaptive Server が起動した日付と時刻。
CountersCleared	datetime		モニタ・カウンタが最後にクリアされた日付と時刻。

monStatementCache

説明

ステートメント・キャッシュについての統計情報を提供します。
monStatementCache テーブルでデータを収集するには、ステートメント・キャッシュを有効にする必要があります。

注意 このモニタリング・テーブルでデータを収集するには、enable monitoring、および enable stmt cache monitoring の設定パラメータを有効にし、statement cache size パラメータを 0 より大きい値に設定してください。

カラム

monStatementCache のカラムは次のとおりです。

名前	タイプ	属性	説明
InstanceID	tinyint		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
TotalSizeKB	int		設定済みのステートメント・キャッシュ・サイズ (KB 単位)。
UsedSizeKB	int		現在使用されているステートメント・キャッシュの量 (KB 単位)。
NumStatements	int		ステートメント・キャッシュに含まれている文の数。
NumSearches	int	カウンタ、リセット	ステートメント・キャッシュが検索された回数。
HitCount	int	カウンタ、リセット	ステートメント・キャッシュが検索され、検索条件と一致する項目が見つかった回数。
NumInserts	int	カウンタ、リセット	ステートメント・キャッシュに挿入された文の数。
NumRemovals	int	カウンタ、リセット	ステートメント・キャッシュから文が削除された回数。この値には、明示的なページを使用して削除された文と置換方式によって削除された文が含まれる。
NumRecompilesSchemaChanges	int	カウンタ、リセット	キャッシュ済みの文で参照されているテーブルでのスキーマ変更起因する再コンパイルの数。
NumRecompilesPlanFlushes	int	カウンタ、リセット	キャッシュからプランがフラッシュされたことに起因する再コンパイルの数。

monSysLoad

説明 環境に固有のテーブル — エンジン単位の統計情報の傾向を提供します。このモニタリング・テーブルにクエリを実行するために `mon_role` の役割は必要ありません。

統計情報ごとに 1 つのエンジンに対して 1 つのローがあります。ただし、例外として `kernel run queue length` はエンジン番号 0 に対してのみレポートされます。

平均は、瞬間的な山と谷を除外するアルゴリズムを使用して計算され、全体的な傾向を示します。

`monSysLoad` に必要な設定パラメータはありません。

コラム `monSysLoad` のコラムは次のとおりです。

名前	データ型	属性	説明
InstanceID	tinyint		クラスタ内でのインスタンスの ID。
EngineNumber	smallint		このローが属するエンジン。
SteadyState	real		Adaptive Server が起動してからのこの統計の平均値。
Avg_1min	real		この統計の 1 分間の移動平均。
Avg_5min	real		この統計の 5 分間の移動平均。
Avg_15min	real		この統計の 15 分間の移動平均。
Max_1min	real		起動時からの 1 分間最大平均。
Max_5min	real		起動時からの 5 分間最大平均。
Max_15min	real		起動時からの 15 分間最大平均。
Max_1min_Time	datetime		Max_1min が発生した日時。
Max_5min_Time	datetime		Max_5min が発生した日時。
Max_15min_Time	datetime		Max_15min が発生した日時。
Statistic			このローが表す次の統計の名前。 <ul style="list-style-type: none"> パーセントの使用率で CPU がビジー パーセントの使用率で I/O がビジー 実行キューの長さ カーネル実行キューの長さ 未処理のまま残っているディスク I/O 1 秒あたりのディスク I/O 1 秒あたりのネットワーク I/O
Sample	float		最後のサンプル間隔の測定基準の値 (現在の測定基準の値)。
Peak	float		インスタンスが起動してからの Sample の最大値 (Sample ピーク値)。
Peak_time	datetime		Peak 値に達した日付と時刻。
StatisticID	int		この統計の固定識別子。ローカライズされた Statistic 名ではなく、固定の StatisticID にアプリケーションを書き込むこともできる。

monSysPlanText

説明

最近実行されたクエリのクエリ・プラン履歴を提供します。

monSysPlanText は、実行中のクエリ・プランの各行から 1 ローのテキストを返します (sp_showplan または set showplan on の戻り値に類似)。monSysPlanText がクエリ・プラン・テキストを適切な順序で読み込むようにするには、クエリ結果を SequenceNumber に基づいて並べ替えます。複数のクエリまたはプロセスのデータを返すクエリでは、クエリ結果を SPID、KPID、BatchID、SequenceNumber に基づいて並べ替えます。

注意 このモニタリング・テーブルでデータを収集するには、enable monitoring、plan text pipe active、および plan text pipe max messages の各設定パラメータを有効にしてください。

カラム

monSysPlanText のカラムは次のとおりです。

名前	データ型	属性	説明
PlanID	int		プランのユニークな識別子。
InstanceID	int		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
SPID	smallint		セッション・プロセス識別子。
KPID	int		カーネル・プロセス識別子。
BatchID	int		プランを生成した SQL バッチのユニークな識別子。
ContextID	int		プロシージャのスタック・フレーム (プロシージャの場合)。
SequenceNumber	int		プラン・テキスト全体の中で PlanText カラムの位置を示す、単調増加する値。
DBID	int		プロシージャが格納されたデータベースのユニークな識別子 (プランがストアド・プロシージャ用の場合)。
ProcedureID	int		プロシージャのユニークな識別子 (プランがストアド・プロシージャ用の場合)。
DBName	varchar(30)	Null	このプランが表す文が実行されるデータベースの名前。monSysPlanText にクエリを実行したときこのデータベースがオープンでない場合、このカラムは NULL。プロセスがストアド・プロシージャまたはその他のコンパイル済みオブジェクトを実行している場合、データベース名は、そのオブジェクトのデータベースの名前になる。
PlanText	varchar(160)	Null	プラン・テキスト出力。

注意 通常、このテーブルには各クエリ・プランの複数のローが含まれます。SequenceNumber カラムを昇順に並べ替えてローを整理してください。

monSysPlanText は、履歴モニタリング・テーブルです。『パフォーマンス&チューニング・シリーズ: モニタリング・テーブル』の『モニタリング・テーブルの概要』の章の「ステートフルな履歴モニタリング・テーブル」を参照してください。

monSysSQLText

説明

最も最近実行された SQL テキスト、または現在実行中の SQL テキストを提供します。返されるローの最大数は、`sql text pipe max messages` で調整できます。

注意 このモニタリング・テーブルでデータを収集するには、`enable monitoring`、`SQL text pipe active`、`max SQL text monitored`、および `SQL text pipe max messages` の各設定パラメータを有効にしてください。

`monSysSQLText` は、履歴モニタリング・テーブルです。『パフォーマンス & チューニング・シリーズ：モニタリング・テーブル』を参照してください。

カラム

`monSysSQLText` のカラムは次のとおりです。

名前	データ型	属性	説明
SPID	smallint		セッション・プロセス識別子。
InstanceID	int		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
KPID	int		カーネル・プロセス識別子。
ServerUserID	int		この SQL テキストを実行したユーザのサーバ・ユーザ識別子 (SUID)。ServerUserID の値は、 <code>syslogins.suid</code> の値に一致する。対応する名前を取得するには <code>suser_name</code> 関数を使用する。
BatchID	int		SQL テキストを含む SQL バッチのユニークな識別子。
SequenceInBatch	int		この SQL テキスト部分のバッチ内での位置を示す (バッチの SQL テキストは複数のローに渡る場合がある)。
SQLText	varchar(255)	Null	SQL テキスト。

注意 多くの場合、クエリのテキストは、このテーブルの複数のローに渡ります。`SequenceInBatch` カラムを昇順に並べ替えてローを適切な順序に整理してください。

monSysStatement

説明 サーバ上で最も最近実行された文の履歴を提供します。statement pipe max messages を使用すると、返される文の統計の最大数を調整できます。

注意 このモニタリング・テーブルでデータを収集するには、enable monitoring、per object statistics active、statement pipe active、statement pipe max messages、および statement statistics active の各設定パラメータを有効にしてください。

monSysStatement は、履歴モニタリング・テーブルです。『パフォーマンス&チューニング・シリーズ：モニタリング・テーブル』を参照してください。

コラム monSysStatements のコラムは次のとおりです。

名前	データ型	属性	説明
SPID	smallint		セッション・プロセス識別子。
InstanceID	int		クラスタ環境に固有のコラム - 共有ディスク・クラスタ内のインスタンスの ID。
KPID	int		カーネル・プロセス識別子。
DBID	int		データベースのユニークな識別子。
ProcedureID	int		プロシージャのユニークな識別子。
PlanID	int		プロシージャの格納されたプランのユニークな識別子。
BatchID	int		文を含む SQL バッチのユニークな識別子。
ContextID	int		プロシージャのスタック・フレーム (プロシージャの場合)。
LineNumber	int		SQL バッチ内の文の行番号。
CpuTime	int	カウンタ	文に使用された CPU 時間 (ミリ秒単位)。
WaitTime	int	カウンタ	タスクが文の実行中に待機した時間 (ミリ秒単位)。
MemUsageKB	int		文の実行に使用したメモリ量 (キロバイト単位)。
PhysicalReads	int	カウンタ	ディスクから読み込まれたバッファ数。
LogicalReads	int	カウンタ	キャッシュから読み込まれたバッファ数。
PagesModified	int	カウンタ	文によって修正されたページ数。
PacketsSent	int	カウンタ	Adaptive Server が送信したネットワーク・パケット数。
PacketsReceived	int	カウンタ	Adaptive Server が受信したネットワーク・パケット数。
NetworkPacketSize	int		セッションで現在設定されているネットワーク・パケットのサイズ (バイト単位)。
PlansAltered	int	カウンタ	実行時に変更されたプラン数。
RowsAffected	int		現在の文の影響を受けたローの数。非効率なクエリ・プランを使用しているクエリでは、多くの場合、返されるローごとの論理 I/O の数値が高い。
ErrorStatus	int		文のエラー・リターン・ステータス。
HashKey	int		文のテキストのハッシュ値。ユニークな識別子ではない。文がステートメント・キャッシュから実行されない場合、このコラムはゼロ (0)。

名前	データ型	属性	説明
SsqlId	int		ステートメント・キャッシュ内でのこの文のクエリ・プランのID。文がステートメント・キャッシュから実行されない場合、このカラムはゼロ (0)。
ProcNestLevel	int		文のネスト・レベル。文がアドホック・クエリである場合、このカラムはゼロ (0)。文がストアド・プロシージャ内にある場合、このカラムは、そのストアド・プロシージャのネスト・レベルを示す。
StatementNumber	int		文がこのプロセスの SQL バッチ内で実行された順番を示す番号。
DBName	varchar(30)		文が実行されるデータベースの名前。monSysStatement にクエリを実行したときこのデータベースがオープンでない場合、このカラムは NULL。プロセスがストアド・プロシージャまたはその他のコンパイル済みオブジェクトを実行している場合、データベース名は、そのオブジェクトのデータベースの名前になる。
StartTime	datetime	Null	文の実行が開始された日付。
EndTime	datetime	Null	文の実行が終了した日付。

monSysWaits

説明 プロセスが待機しているイベントに関するサーバワイドの統計情報を提供します。

注意 このモニタリング・テーブルでデータを収集するには、**enable monitoring** および **wait event timing** の各設定パラメータを有効にしてください。

カラム monSysWaits のカラムは次のとおりです。

名前	データ型	属性	説明
InstanceID	int		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
WaitEventID	smallint		待機イベントのユニークな識別子。
WaitTime	int	カウンタ	タスクがイベントを待機するために費やした時間 (秒単位)。
Waits	int		タスクがイベントを待機した回数。

詳細については、『パフォーマンス&チューニング・シリーズ：モニタリング・テーブル』を参照してください。

WaitEventID カラムをジョイン・カラムとして使用して **monSysWaits** テーブルを **monWaitEventInfo** にジョインすると、待機イベントの説明を取得できます。次に例を示します。

```
select w.Waits, w.WaitTime, w.WaitEventID, i.Description
from master..monSysWaits w, master..monWaitEventInfo i
where w.WaitEventID = i.WaitEventID
```

monSysWorkerThread

説明 ワーカー・スレッドの設定と実行に関するサーバワイドの統計情報を返します。

注意 このモニタリング・テーブルでデータを収集するには、`enable monitoring` 設定パラメータを有効にしてください。

カラム monSysWorkerThread のカラムは次のとおりです。

名前	データ型	属性	説明
InstanceID	int		クラスタ環境に固有のカラム - 共有ディスク・クラスタ内のインスタンスの ID。
ThreadsActive	int		現在アクティブなワーカー・プロセス数。
TotalWorkerThreads	int		ワーカー・プロセスの最大数 (number of worker processes を使用して設定)。
HighWater	int	リセット	これまでに使用されたワーカー・プロセスの最大数。
ParallelQueries	int	カウンタ、リセット	試行された並列クエリの数。
PlansAltered	int	カウンタ、リセット	ワーカー・プロセスが使用できなかったために変更されたプラン数。
WorkerMemory	int		ワーカー・プロセスが現在使用しているメモリ量。
TotalWorkerMemory	int		ワーカー・プロセスが使用できる設定メモリ量。
WorkerMemoryHWM	int	リセット	ワーカー・プロセスがこれまでに使用した最大メモリ量。
MaxParallelDegree	int		使用可能な最大並列度。max parallel degree 設定オプションの現在の Run Value。
MaxScanParallelDegree	int		スキャンに使用可能な最大並列度。max scan parallel degree 設定オプションの現在の Run Value。

monTableColumns

説明

各モニタリング・テーブルのすべてのカラムを説明します。monTableColumnsを使用すると、モニタリング・テーブルにあるカラムを調べることができます。monTableColumnsをmonTablesにジョインすると、モニタリング・テーブルのカラムとカラム属性のレポートが得られます。

このテーブルのメタデータ・ビューは、共有ディスク・クラスタ内のすべてのインスタンスで同じです。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム

monTableColumnsのカラムは次のとおりです。

名前	データ型	属性	説明
TableID	int		ビューのユニークな識別子。
ColumnID	int		カラムの位置。
TypeID	int		カラムのデータ型を示す識別子。
Precision	tinyint		カラムの精度 (数値の場合)。
Scale	tinyint		カラムの位取り (数値の場合)。
Length	smallint		カラムの最大長 (バイト数)。
Indicators	int		カラムの特定のプロパティを示すインジケータ (たとえば、カラムが循環性であり、サンプリングが必要である場合) ¹ 。
TableName	varchar(30)	Null	テーブルの名前。
ColumnName	varchar(30)	Null	カラムの名前。
TypeName	varchar(20)	Null	カラムのデータ型の名前。
Description	varchar(255)	Null	カラムの説明。

¹Indicators カラムはビットマップです。ビット・マスクを使用すると、どのビットがオンかを判断できます。有効な値は次のとおりです。

- 1 – Indicators の値が急速に増加し、 2^{32} に達するとカウンタの循環が発生する場合があります。これは、Indicators カラム値の 1 ビットがオンになっているカラムで発生する可能性があります。1 ビットがオンかどうかを判断するには、以下を使用します。

```
select TableName, ColumnName
from Master..monTableColumns
where Indicators & 1 != 0
```

- 2 – カウンタが sp_sysmon と共有されており、sp_sysmon.. clear を実行するとリセットされます。sp_sysmon が clear パラメータを使用してクリアするすべてのカラムを表示するには、以下を使用します。

```
Select TableName, ColumnName
from master..monTableColumns
where Indicators & 2 != 0
```

monTableParameters

説明

モニタリング・テーブルのクエリ・パフォーマンスを最適化するために使用する、モニタリング・テーブル内のすべてのカラムについての説明を提供します。このテーブルのメタデータ・ビューは、共有ディスク・クラスタ内のすべてのインスタンスで同じです。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム

monTableParameters のカラムは次のとおりです。

名前	データ型	属性	説明
TableID	int		テーブルのユニークな識別子。
ParameterID	int		パラメータの位置。
TypeID	int		パラメータのデータ型を示す識別子。
Precision	tiny_int		パラメータの精度 (数値の場合)。
Scale	tiny_int		パラメータの位取り (数値の場合)。
Length	small_int		パラメータの最大長 (バイト数)。
TableName	varchar(30)	Null	テーブルの名前。
ParameterName	varchar(30)	Null	パラメータの名前。
TypeName	varchar(20)	Null	パラメータのデータ型の名前。
Description	varchar(255)	Null	パラメータの説明。

monTables

説明

すべてのモニタリング・テーブルの説明を提供します。monTables を monTableColumns にジョインすると、各モニタリング・テーブルとそのカラムの説明が得られます。

このテーブルのメタデータ・ビューは、共有ディスク・クラスタ内のすべてのインスタンスで同じです。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム

monTables のカラムは次のとおりです。

名前	データ型	属性	説明
TableID	int		テーブルのユニークな識別子。
カラム	tinyint		テーブルの総カラム数。
Parameters	tinyint		指定可能なオプション・パラメータの総数。
Indicators	int		テーブルの特定のプロパティを示すインジケータ (たとえば、テーブルがセッション・コンテキストを保持する場合) ¹ 。
Size	int		最大ロー・サイズ (バイト数)。
TableName	varchar(30)	Null	テーブル名。
Description	varchar(368)	Null	テーブルの説明。

¹Indicators カラムはビットマップです。ビット・マスクを使用すると、どのビットがオンかを判断できます。値が1の場合、テーブルは履歴テーブルです。

すべての履歴テーブルを表示するには、以下を実行します。

```
Select TableName
from master..monTables
where Indicators & 1 != 0
```

monTableTransfer

説明

MonTableTransfer は、Adaptive Server のアクティブなメモリに存在するテーブルの転送履歴情報を提供します。完了済みの転送に関する情報は保管されません。MonTableTransfer は、増分転送のマークが付いているかどうかにかかわらず、すべてのテーブルの現在進行中の転送に関する情報を提供します。また、増分転送のマークが付いているテーブルの以前の転送に関する情報も提供します。

カラム

monTableTransfer のカラムは次のとおりです。

名前	データ型	属性	説明
InstanceID	tinyint	NOT NULL	クラスタ環境に固有のカラム - コマンドを実行中のサーバのインスタンス ID を保持する。ノンクラスタード・サーバでは、常にゼロ。
DBID	smallint		テーブルのデータベース ID。
TableID	int		テーブルのユニークな識別子。
TableName	varchar(255)	NULL	テーブルの名前。
SequenceID	int		Adaptive Server で生成される内部トラッキング ID。
TrackingID	int	NULL	ユーザが指定するトラッキング ID。
PercentDone	smallint		0 ~ 100 の整数で表される転送作業終了率 (完了した転送を 100 と表示)。
BeginTime	datetime		転送が開始された日付と時刻。
EndTime	datetime	NULL	転送が終了した日付と時刻。進行中の転送は NULL。
EndCode	smallint	NULL	転送の終了ステータス。 <ul style="list-style-type: none"> • 0 - 正常に終了した転送 • NULL - 進行中の転送 • エラー・コード - 失敗した転送
TransferFloor	bigint		データを送信できる時点のタイムスタンプ。
TransferCeiling	bigint		データがコミットされておらず、送信できない時点のタイムスタンプ。
RowsSent	bigint		送信したローの数。
BytesSent	bigint		送信したバイト数。
Format	varchar(8)	NOT NULL	送信先フォーマットの名前。ase、bcp、csv、または iq のいずれか。

monTempdbActivity

説明 クラスタ環境に固有のテーブル。インスタンスが `tempdb` 設定モードで開始された場合、グローバル・システム `tempdb` も含めてオープンしているすべてのローカル・テンポラリ・データベースの統計情報を提供します。

`monTempdbActivity` でデータを収集するには、`enable monitoring`、`per object statistics active`、および `object lockwait timing` の各設定パラメータを有効にする必要があります。

カラム `monTempdbActivity` のカラムは次のとおりです。

名前	データ型	説明
DBID	int	データベースのユニークな識別子。
InstanceID	tinyint	クラスタ内でのインスタンスの ID。
DBName	varchar(30)	データベースの名前。
AppendLogRequest	int	データベース・トランザクション・ログへの追加を試行しているインスタンスからのセマフォ要求の数。
AppendLogWaits	int	ログ追加セマフォが付与されるまでタスクが待機する回数。
LogicalReads	int	読み込まれたバッファの総数。
PhysicalReads	int	ディスクから読み込まれたバッファ数。
APFReads	int	読み込まれた非同期プリフェッチ (APF) バッファの数。
PagesRead	int	読み込まれたページの総数。
PhysicalWrites	int	ディスクに書き込まれたバッファの総数。
PagesWritten	int	ディスクに書き込まれたページの総数。
LockRequests	int	このテンポラリ・データベース内のオブジェクト・ロック要求の数。
LockWaits	int	このテンポラリ・データベース内でタスクがオブジェクト・ロックを待機した回数。
CatLockRequests	int	システム・カタログ上のロック要求の数。
CatLockWaits	int	タスクがシステム・テーブルのロックを待機した回数。
AssignedCnt	int	このテンポラリ・データベースがユーザ・タスクに割り付けられた回数。
SharableTabCnt	int	作成された共有可能なテーブルの数。

monWaitClassInfo

説明

すべての待機クラスの説明をテキストで提供します(たとえば、ディスクの読み込み完了の待機など)。すべての待機イベント (`monWaitEventInfo` の説明を参照) は、プロセスが待機するイベントのタイプを分類した、さまざまな待機クラスにグループ化されています。

このテーブルは、共有ディスク・クラスタ内のすべてのインスタンスで同じ情報を表示します。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム

`monWaitClassInfo` のカラムは次のとおりです。

名前	データ型	属性	説明
WaitClassID	smallint		待機イベント・クラスのユニークな識別子。
Description	varchar(50)	Null	待機イベント・クラスの説明。

monWaitEventInfo

説明 Adaptive Server でプロセスが待機する可能性のあるあらゆる状況について、説明をテキストで提供します。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム monWaitEventInfo のカラムは次のとおりです。

名前	データ型	属性	説明
WaitEventID	smallint		待機イベントのタイプを示すユニークな識別子。
WaitClassID	smallint		待機イベント・クラスのユニークな識別子。
Description	varchar(50)	Null	待機イベント・タイプの説明。

WaitEventID カラムで monWaitEventInfo を monProcessWaits または monSysWaits にジョインすると、これらのテーブルにリストされた待機イベントの説明を取得できます。

monWorkload

説明

クラスタ環境に固有のテーブル — インスタンスごとに各論理クラスタの負荷スコアをその負荷プロファイルに従って表示します。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム

monWorkload のカラムは次のとおりです。

名前	データ型	属性	説明
LCID	tinyint		論理クラスタ ID。
InstanceID	tinyint		クラスタ内でのインスタンスの ID。
LoadProfileID	tinyint		負荷スコアを生成するために使用された負荷プロファイルの ID。
LoadScore	int		このインスタンスまたは論理クラスタの負荷スコア。
ConnectionsScore	float		user connections 測定基準の重み付けされた値。
CpuScore	float		cpu utilization 測定基準の重み付けされた値。
RunQueueScore	float		run queue 測定基準の重み付けされた値。
IoLoadScore	float		io load 測定基準の重み付けされた値。
EngineScore	float		engine deficit 測定基準の重み付けされた値。
UserScore	float		user 測定基準の重み付けされた値。
LogicalClusterName	varchar(30)		論理クラスタ名。
InstanceName	varchar(30)		インスタンス名。
LoadProfileName	tinyint		負荷スコアを生成するために使用された負荷プロファイルの名前。

monWorkloadPreview

説明

クラスタ環境に固有のテーブル — プロファイルを有効にせずに負荷プロファイルが負荷スコアに及ぼす影響の度合いの評価を提供します。monWorkloadPreviewには、論理クラスタが実行されている論理クラスタおよびインスタンスごとに1つのローが含まれます。負荷スコアとコンポーネントは、その論理クラスタの現在のプロファイルに基づいています。monWorkloadPreview テーブルには、システム上で設定されているインスタンスと負荷プロファイルの組み合わせごとに1つのローがあります。これによって、管理者は、各プロファイルで負荷スコアがどのように計算されるかを確認できます。このモニタリング・テーブルにクエリを実行するために mon_role の役割は必要ありません。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム

monWorkloadPreview のカラムは次のとおりです。

名前	データ型	属性	説明
InstanceID	tinyint		クラスタ内でのインスタンスの ID。
LoadProfileID	smallint		負荷プロファイル ID。
LoadScore	int		このインスタンスまたは論理クラスタの負荷スコア。
ConnectionScore	float		user connections 測定基準の重み付けされた値。
CpuScore	float		cpu utilization 測定基準の重み付けされた値。
RunQueueScore	float		run queue 測定基準の重み付けされた値。
IoLoadScore	float		io load 測定基準の重み付けされた値。
EngineScore	float		engine deficit 測定基準の重み付けされた値。
UserScore	float		user 測定基準の重み付けされた値。
InstanceName	varchar(30)		インスタンス名。
LoadProfileName	varchar(30)		負荷スコアを生成するために使用された負荷プロファイルの名前。

monWorkloadProfile

説明 クラスタ環境に固有のテーブル – 現在設定されている負荷プロファイルを表示します。このモニタリング・テーブルにクエリを実行するために `mon_role` の役割は必要ありません。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

コラム monWorkloadProfile のコラムは次のとおりです。

名前	データ型	属性	説明
ProfileID	smallint		負荷プロファイル ID。
ConnectionsWeight	tinyint		active connections 測定基準に関連付けられている重み付け。
CpuWeight	tinyint		cpu utilization 測定基準に関連付けられている重み付け。
RunQueueWeight	tinyint		run queue 測定基準に関連付けられている重み付け。
IoLoadWeight	tinyint		io load 測定基準に関連付けられている重み付け。
EngineWeight	tinyint		engine deficit 測定基準に関連付けられている重み付け。
UserWeight	tinyint		user metric 測定基準に関連付けられている重み付け。
LoginThreshold			ログイン負荷分散のスレッシュホールド。
DynamicThreshold	smallint		動的負荷分散のスレッシュホールド (つまり、負荷を目的としたログイン後マイグレーション)。
Hysteresis	tinyint		リダイレクトを有効にする最小負荷スコア。
Name	varchar(30)		負荷プロファイル名。
Type	varchar(30)		負荷プロファイルのタイプ。プロファイルがユーザ定義か、システム定義かを示す。値は次のとおり。 <ul style="list-style-type: none"> • User • System

monWorkloadRaw

説明

クラスタ環境に固有のテーブル — インスタンスごとにロー負荷統計情報を提供します。このモニタリング・テーブルにクエリを実行するために `mon_role` の役割は必要ありません。

注意 このモニタリング・テーブルでデータを収集するために、有効にする必要のある設定パラメータはありません。

カラム

`monWorkloadRaw` のカラムは次のとおりです。

名前	データ型	属性	説明
InstanceID	tinyint		クラスタ内でのインスタンスの ID。
ConnectionsRaw	float		<code>user connections</code> 測定基準のロー値。
CpuRaw	float		<code>cpu utilization</code> 測定基準のロー値。
RunQueueRaw	float		<code>run queue</code> 測定基準のロー値。
IoLoadRaw	float		<code>io load</code> 測定基準のロー値。
EngineRaw	float		<code>engine deficit</code> 測定基準のロー値。
UserRaw	float		<code>user</code> 測定基準のロー値。
InstanceName	varchar(30)		インスタンス名。

sybpcidb のテーブル

sybpcidb データベースには、Java PCI Bridge および PCA/JVM プラグインの設定情報が格納されます。この章では、sybpcidb のテーブルをアルファベット順に説明します。

Java 用にサーバを設定するときは、sybpcidb を作成して、そのテーブルをインストールし、そのシステム・ストアド・プロシージャを作成します。詳細については、プラットフォームのインストール・ガイドを参照してください。また、sybpcidb での設定と情報の表示のために使用する sp_jreconfig ストアド・プロシージャおよび sp_pciconfig ストアド・プロシージャの使用法の詳細については、『Adaptive Server Enterprise における Java』も参照してください。

pca_jre_arguments

説明 PCA/JVM プラグインを設定するために使用される引数に関する情報が格納されています。

カラム sybpcidb に配置されています。pca_jre_arguments のカラムは次のとおりです。

名前	データ型	説明
jre_args_directive_index	int	引数が属するディレクティブのインデックス。
jre_args_name	varchar(255)	引数の名前。
jre_args_units	varchar(255)	引数のタイプ。値は次のとおり。 <ul style="list-style-type: none"> • switch • string • number • array
jre_args_number_value	int	units=number の場合に、引数に関連付けられている数値を保持する。
jre_args_string_value	varchar(255)	units=string または units=array の場合、引数に関連付けられている文字列値を保持する。
jre_args_description	varchar(255)	テキストによる引数の簡単な説明。
jre_args_enabled	int	値は次のとおり。 <ul style="list-style-type: none"> • 0 – 有効でない • 1 – 有効 (デフォルト)
jre_args_status	int	今後のために予約済み。

インデックス

- jre_args_directive_index、jre_args_name、jre_args_string_value にユニーク・クラスタード・インデックスが作成されています。

pca_jre_directives

説明 PCA/JVM の設定に使用されるディレクティブに関する情報が格納されています。

カラム sybpcidb に配置されています。pca_jre_directives のカラムは次のとおりです。

名前	データ型	説明
jre_directives_index	int	ディレクティブのインデックス。
jre_directives_name	varchar(255)	ディレクティブの名前。
jre_directives_description	varchar(255)	テキストによるディレクティブの説明。
jre_directives_enabled	int	値は次のとおり。 <ul style="list-style-type: none"> • 0 – 有効でない • 1 – 有効 (デフォルト)
jre_directives_status	int	今後のために予約済み。

インデックス

- jre_directives_name にユニーク・クラスタード・インデックスが作成されています。
- jre_directives_index にユニーク・ノンクラスタード・インデックスが作成されています。

pci_arguments

説明 PCI Bridge を設定するために使用される各引数を定義する情報が格納されています。

カラム sybpcidb に配置されています。pci_arguments のカラムは次のとおりです。

名前	データ型	説明
pci_args_directive_index	int	引数が属するディレクティブのインデックス。
pci_args_name	varchar(255)	引数の名前。
pci_args_units	varchar(255)	単位のタイプ。値は次のとおり。 <ul style="list-style-type: none"> • switch • number
pci_args_number_value	int	units=number の場合、数値。units=switch の場合、値はゼロ (0)。
pci_args_string_value	varchar(255)	今後のために予約済み。
pci_args_description	varchar(255)	引数とその目的についてのテキストによる簡単な説明。
pci_args_enabled	int	値は次のとおり。 <ul style="list-style-type: none"> • 0 – 有効でない • 1 – 有効 (デフォルト)
pci_args_status	int	今後のために予約済み。

インデックス

- pci_args_directive_index、pci_args_name にユニーク・クラスタード・インデックスが作成されています。

pci_directives

説明 PCI Bridge を設定するディレクティブが格納されています。

カラム sybpcidb に配置されています。pci_directives のカラムは次のとおりです。

名前	データ型	説明
pci_directives_index	int	ディレクティブのインデックス。
pci_directives_name	varchar(255)	ディレクティブの名前。
pci_directives_description	varchar(255)	ディレクティブの説明。
pci_directives_enabled	int	値は次のとおり。 <ul style="list-style-type: none"> • 0 – 有効でない • 1 – 有効 (デフォルト)
pci_directives_status	int	今後のために予約済み。

インデックス

- pci_directives_name にユニーク・クラスタード・インデックスが作成されています。
- pci_directives_index にユニーク・ノンクラスタード・インデックスが作成されています。

pci_slotinfo

説明 スロットのディレクティブおよび引数のテーブル名など、各スロットに関する情報が含まれています。

カラム sybpcidb に配置されています。pci_slotinfo のカラムは次のとおりです。

名前	データ型	説明
slot_number	int	スロットの数。
slot_name	varchar(255)	JVM など、スロットの名前。
slot_pca_directives_table_name	varchar(255)	pca_jre_directives など、PCA ディレクティブ・テーブルの名前。
slot_pca_arguments_table_name	varchar(255)	pca_jre_arguments など、PCA 引数テーブルの名前。
slot_status	varchar(255)	今後のために予約済み。

インデックス

- slot_name にユニーク・クラスタード・インデックスが作成されています。
- slot_number にユニーク・ノンクラスタード・インデックスが作成されています。

pci_slot_syscalls

説明

カラム sybpcidb に配置されています。pci_slot_syscalls のカラムは次のとおりです。

名前	データ型	説明
syscall_slot_number	int	システム呼び出しに関連付けられているスロット番号。
syscall_system_call	varchar(255)	システム呼び出しの名前。
syscall_dispatch_name	varchar(255)	システム呼び出しのディスパッチ関数の名前。
syscall_enabled	int	値は次のとおり。 <ul style="list-style-type: none"> • 0 – 有効でない • 1 – 有効 (デフォルト)
syscall_status	int	今後のために予約済み。

インデックス

- `syscall_slot_number`、`syscall_system_call` にユニーク・クラスタード・インデックスが作成されています。

索引

記号

- () (カッコ)
 - SQL 文内 xiii
- , (カンマ)
 - SQL 文内 xiv
- ::= (BNF 表記)
 - SQL 文内 xiv
- [] (角カッコ)
 - SQL 文内 xiv
- { } (中カッコ)
 - SQL 文内 xiv

数字

- 2 フェーズ・コミット
 - probe プロセス 51

A

- allow updates to system tables 設定パラメータ 8

B

- Backus Naur Form (BNF) 表記 xiii, xiv
- BNF 表記、SQL 文内 xiii, xiv

D

- dbid カラム、sysusages テーブル 96
- DTX パティシパント 26

E

- ESP。「拡張ストアド・プロシージャ」参照

G

- grant コマンド
 - sysprotects テーブル 69

I

- identity 値 5
- ID、サーバ役割
 - sysroles テーブル 76

J

- Java 項目
 - sysjars テーブル 43
 - sysxtypes テーブル 99

M

- master データベース
 - システム・テーブル 1-2
- monCachedObject テーブル 120
- monCachedProcedures テーブル 122
- monCachedStatement テーブル 123
- monCachePool テーブル 121
- monCIPC テーブル 126
- monCIPCEndpoints テーブル 127
- monCIPCLinks テーブル 128
- monCIPCMesh テーブル 129
- monClusterCacheManager テーブル 133
- monDataCache テーブル 135, 147
- monDeadLock テーブル 139
- monDeviceIO テーブル 141
- monEngine テーブル 142
- monErrorLog テーブル 144
- monIOQueue テーブル 146
- monLicense テーブル 147
- monLocks テーブル 148
- monLogicalCluster テーブル 150
- monLogicalClusterAction テーブル 152

索引

monLogicalClusterRoute 172
monLogicalClusterRoute テーブル 154
monNetworkIO テーブル 155
monOpenDatabases テーブル 156
monOpenObjectActivity テーブル 157, 160
monOpenPartitionActivity テーブル 160
monPCIBridge 163
monPCIEngine 163
monPCISlots 164
monProcedureCache テーブル 167
monProcedureCacheMemoryUsage テーブル 168
monProcedureCacheModuleUsage テーブル 169
monProcess テーブル 170
monProcessActivity テーブル 171
monProcessLookup テーブル 172
monProcessNetIO テーブル 173
monProcessObject テーブル 174
monProcessProcedures テーブル 175
monProcessSQLText テーブル 176
monProcessStatement テーブル 177
monProcessWaits テーブル 178, 179
monProcessWorkerThread テーブル 179
monState テーブル 180
monStatementCache テーブル 181
monSysLoad テーブル 182
monSysPlanText テーブル 183
monSysSQLText テーブル 184
monSysStatement テーブル 185
monSysWaits テーブル 187
monSysWorkerThread テーブル 188
monTableColumns テーブル 189
monTableParameters テーブル 190
monTables テーブル 191
monTableTransfer 192
monTempdbActivity テーブル 193
monWaitClassInfo テーブル 194
monWaitEventInfo テーブル 195
monWorkload テーブル 196
monWorkloadPreview テーブル 197
monWorkloadProfile テーブル 198
monWorkloadRaw テーブル 199

O

OAM (オブジェクト・アロケーション・マップ) ページ数 85

P

probe プロセス、2 フェーズ・コミットの設定 51

R

revoke コマンド
sysprotects テーブル 69

S

segmap カラム、sysusages テーブル 96
segment カラム、syssegments テーブル 78
spid 番号 66
sysaudits テーブル 16
syslogshold 53
suid (サーバ・ユーザ ID)
sysalternates テーブル・リスト 10
syslogins テーブル・リスト 50
sybdiagdb データベース 4
syblicenseslog テーブル 4, 9
sybpcidb データベース
pca_jre_arguments 202
pca_jre_directives 203
pci_arguments 204
pci_directives 205
pci_slot_syscalls 207
pci_slotinfo 206
sybsecurity データベース
システム・テーブル 2
sybsystemdb データベース
システム・テーブル 2
sysalternates テーブル 10
sysaltusages テーブル 11
sysattributes テーブル 13–14
sysauditoptions テーブル 15
sysaudits_01 ~ sysaudits_08 テーブル 16–17
syscharsets システム・テーブルの文字セット 18
syscharsets テーブル 18
syscolumns テーブル 19–20
syscomments テーブル 21–22
sysconfigures テーブル 23–24
sysconstraints テーブル 25
syscoordinations テーブル 26
syscurconfigs テーブル 27
sysdatabases テーブル 29–31
sysdepends テーブル 32
sysdevices テーブル 33–34

sysdevices 内の *status* ビット 33
sysengines テーブル 37
sysgams テーブル 38
sysindexes テーブル 39–41
sysinstances システム・テーブル 42
sysjars テーブル 43
syskeys テーブル 44
syslanguages テーブル 45
syslisteners テーブル 46
syslocks テーブル 47–48
sysloginroles テーブル 49
syslogins テーブル 50–51
syslogs テーブル 52
 無限ループ (変更する場合) 52
syslogshold テーブル 53
sysmessages テーブル 54
sysmonitors テーブル 55
sysobjects テーブル 56–60
syspartitionkeys テーブル 62
sysprocedures テーブル 65
sysprocesses テーブル 66–68
sysprotects テーブル 69
sysquerymetrics テーブル 70
sysqueryplans テーブル 72
sysreferences テーブル 73
sysremotelogins テーブル 74
sysresourcelimits テーブル 75
sysroles テーブル 76
syssecmechs テーブル 77
syssegments テーブル 78
sysservers テーブル 79–80
sysessions テーブル 81
syslices テーブル 82
sysrvroles テーブル 83
sysstatistics テーブル 84
sysabstats テーブル 85–86
systhresholds テーブル 87
sys timeranges テーブル 89
sys transactions テーブル 90–92
sytypes テーブル 93–95
sysusages テーブル 96
sysusermessages テーブル 97
sysusers テーブル 98
 sysalternates テーブル 10
sysxtypes テーブル 99

T

tempdb データベース
 システム・テーブル・エントリ 56–60
timestamp カラム 4

U

us_english 言語 45

あ

アーカイブ・データベースへのアクセス
 sysaltusages テーブル 11
 スクラッチ・データベース 11
 アロケーション・ユニット
 sysusages テーブル 96
 暗号化
 役割パスワード 83
 ユーザ・パスワード 50

い

依存性、データベース・オブジェクト
 sysdepends テーブル 32
 インデックス
 システム・テーブル・エントリ 39

え

英語、U.S。「*us_english* 言語」参照
 エイリアス、言語
 syslanguages テーブル 45
 エイリアス、ユーザ
 sysalternates テーブル 10
 エラー・メッセージ
 システム・テーブル・エントリ 54
 エンジン
 sysengines テーブル 37
 システム・テーブル・エントリ 37

索引

お

大文字と小文字の区別
SQL xv

か

階層

データ型 93

外部キー

syskeys テーブル 44

角カッコ []

SQL 文内 xiv

角カッコ。「角カッコ []」参照

拡張ストアド・プロシージャ

システム・テーブル・エントリ 21, 56-60

カッコ ()

SQL 文内 xiii

月の値

syslanguages テーブル 45

代替言語 45

短縮形 (省略形) 45

カラム

予約済み 8

監査

sysauditoptions テーブル 15

sysaudits_01 ~ *sysaudits_08* テーブル 16, 202

カンマ (,)

SQL 文内 xiv

き

キー、テーブル

syskeys テーブル 44

記号

SQL 文内 xiii

規則

Transact-SQL の構文 xiii

リファレンス・マニュアル xiii

「構文」参照

共通キー

syskeys テーブル 44

く

クライアント

フェールバック中の削除 81

グループ

sysusers テーブル・エントリ 98

グローバル・アロケーション・マップのページ 38

グローバル監査オプション、*sysauditoptions* システム・
テーブル 15

け

言語、代替

syslanguages テーブル 45

システム・テーブル・エントリ 45

検索

オブジェクトの依存性 32

オブジェクトの定義 21, 65

言語 45

制約 25

設定パラメータ 23, 27

データ型 93

データベース ID 29

データベース・オブジェクト 56

データベース設定 29

データベース内のユーザ 98

データベース名 29

デバイス名 33

パーミッション情報 69

文字セット 18

役割 76

ユーザ・エイリアス 10

リソース制限 75

検査制約

sysconstraints テーブル 25

システム・テーブル・エントリ 56-60, 65

こ

- 高可用性
 - 再接続情報 81
- 更新
 - システム・テーブル 8
 - システム・テーブルの直接更新 8
 - システム・プロシージャ 8
- 構造
 - 設定 27
- 構文規則、Transact-SQL xiii

さ

- サイズ
 - ロー 85
- 作業領域
 - 削除 102
- 削除
 - 作業領域 102
- 削除されたロー
 - 数 85
- 参照整合性制約
 - sysconstraints* テーブル 25
 - sysobjects* テーブル 56-60
 - sysreferences* テーブル 73

し

- シーケンス・ツリー、オブジェクト 65
- 時間範囲
 - syntimeranges* システム・テーブル 89
- システム・テーブル 1-9
 - allow updates to system tables* パラメータ 8
 - master* データベース 1-2
 - キー 44
 - 更新 8
 - 直接更新 8
 - トリガ 8
 - パーミッション 7
- システム・プロシージャ
 - 更新 8
- システム役割
 - sysloginroles* テーブル 49
 - sysrvroles* テーブル 83
- 情報 (サーバ)
 - 設定パラメータ 23, 27
 - データベース 29-31

す

- 数 (量)
 - OAM ページ 85
 - インデックス・リーフ・ページ 85
 - インデックス・レベル 85
 - 削除されたロー 85
 - 転送されたロー 85
 - ページ 85
 - ロー 85
- スクラッチ・データベース 11
- ストアド・プロシージャ
 - オブジェクトの依存性 32
 - システム・テーブル・エントリ 21, 56-60, 65
- スライス
 - システム・テーブル・エントリ 82
- スレシヨルド
 - systrsholds* テーブル 87

せ

- 制約
 - sysconstraints* テーブル 25
 - sysreferences* テーブル 73
 - システム・テーブル・エントリ 21, 56-60
- セグメント
 - syssegments* テーブル 78
 - syslices* テーブル 82
 - システム・テーブル・エントリ 78
- 設定パラメータ
 - システム・テーブル 23, 27

そ

- ソート順
 - syscharsets* システム・テーブル 18

た

- ダンプ・デバイス
 - sysdevices* テーブル 33
 - システム・テーブル・エントリ 33

ち

- 中カッコ {}, SQL 文内 xiv
- 直接更新
 - システム・テーブル 8

て

- ディスク・デバイス
 - sysdevices* テーブル 33
- ディスク・ミラーリング
 - sysdevices* テーブルの status 34
- ディスク割り付けの集まり 96
- データ型
 - systypes* テーブル 93–95
 - 階層 93
 - リスト 93
- データベース
 - システム・テーブル・エントリ 29
- データベース・オブジェクト
 - sysobjects* テーブル 56–60
 - 依存性 32
- データベース・デバイス
 - sysdevices* テーブル 33
 - システム・テーブル・エントリ 33
- データ・ロー
 - サイズ 85
- テープ・ダンプ・デバイス
 - sysdevices* テーブル 33
- テーブル
 - monCachedObject* 120
 - monCachedProcedures* テーブル 122
 - monCachePool* 121
 - monCIPC* 126
 - monCIPCEndpoints* 127
 - monCIPCLinks* 128
 - monCIPMesh* 129
 - monClusterCacheManager* 133
 - monDataCacher* 135
 - monDeadlLock* 139
 - monDeviceIO* 141
 - monEngine* 142
 - monErrorLog* 144
 - monIOQueue* 146
 - monLicense* 147
 - monLocks* 148
 - monLogicalCluster* 150
 - monLogicalClusterAction* 152
 - monLogicalClusterRoute* 154
 - monNetworkIO* 155
 - monOpenDatabases* 156
 - monOpenObjectActivity* 157
 - monOpenPartitionActivity* 160
 - monProcedureCache* 167
 - monProcedureCacheMemoryUsage* 168
 - monProcedureCacheModuleUsage* 169
 - monProcess* 170
 - monProcessActivity* 171
 - monProcessLookup* 172
 - monProcessNetIO* 173
 - monProcessObject* 174
 - monProcessProcedures* 175
 - monProcessSQLText* 176
 - monProcessStatement* 177
 - monProcessWaits* 178
 - monProcessWorkerThread* 179
 - monState* 180
 - monStatementCache* 181
 - monSysLoad* 182
 - monSysPlanText* 183
 - monSysSQLText* 184
 - monSysStatement* 185
 - monSysWaits* 187
 - monSysWorkerThread* 188
 - monTableColumns* 189
 - monTableParameters* 190
 - monTables* 191
 - monTempdbActivity* 193
 - monWaitClassInfo* 194
 - monWaitEventInfo* 195
 - monWorkload* 196
 - monWorkloadPreview* 197
 - monWorkloadProfile* 198
 - monWorkloadRaw* 199
 - オブジェクトの依存性 32
 - システム・テーブル・エントリ 19, 56–60
- デバイス
 - システム・テーブル・エントリ 33
- デフォルト
 - システム・テーブル・エントリ 21, 56–60, 65
- 転送されたロー
 - 数 85

と

- 統計値
 - システム・テーブル 84, 85
- トランザクション 90
- トランザクション・ログ
 - システム・テーブル・エントリ 56–60
- トリガ
 - オブジェクトの依存性 32
 - システム・テーブル 8
 - システム・テーブル・エントリ 21, 56–60, 65

な

- 名前
 - ソート順 18
 - 文字セット 18
- 名前付き時間範囲
 - systemranges* システム・テーブル 89

に

- 偽のテーブルのマテリアライズ 5

は

- パーミッション
 - sysprotects* テーブル 69
 - システム・テーブル 7
 - システム・テーブル・エントリ 69

ひ

- ビュー
 - オブジェクトの依存性 32
 - システム・テーブル・エントリ 19, 21, 56–60, 65

ふ

- プライマリ・キー
 - syskeys* テーブル 44
- プラン
 - オブジェクト 65
- プロセス (サーバのタスク)
 - sysprocesses* テーブル 66
 - システム・テーブル・エントリ 66
- プロセスのブロック 66
- 分散トランザクション管理 (DTM) 26

へ

- ページ、インデックス
 - 数 85
- ページ、グローバル・アロケーション・マップ 38
- ページ、データ
 - 数 85

ま

- マッピング
 - sysusages* テーブル 96

め

- メッセージ
 - sysmessages* テーブル 54
 - sysusermessages* テーブル 97
 - ユーザ定義 97

も

- モニタリング・テーブル 119–175
 - monLogicalClusterRoute* 172
 - リストおよび説明、モニタリング・テーブル 119

索引

や

役割

sysroles テーブル 76

sysrrvroles テーブル 83

役割、システム

sysloginroles テーブル 49

ゆ

ユーザ

syslogins テーブル 50-51

sysusers テーブル 98

システム・テーブル・エントリ 50-51, 98

ユーザ定義役割

sysrrvroles テーブル 83

よ

予約済みカラム 8

り

リスト

システム・テーブル 1-4

リソース制限

sysresourcelimits テーブル 75

リファレンス情報

dbcc テーブル 101

システム・テーブル 3

リモート・サーバ

syssservers テーブル 79

システム・テーブル・エントリ 79

リモート・プロシージャ・コール

sysremotelogins テーブル 74

syssservers テーブル 79

リモート・ログイン

sysremotelogins テーブル 74

システム・テーブル・エントリ 74

領域の割り付け

sysusages テーブル 96

システム・テーブル・エントリ 96

る

ループ

syslogs の変更と無限ループ 52

ルール

システム・テーブル・エントリ 21, 56-60, 65

ろ

ロー、インデックス

サイズ 85

リーフのサイズ 85

ロー、データ

数 85

ロー、テーブル

サイズ 85

ログイン

“probe” 51

syslogins テーブル 50-51

sysremotelogins テーブル 74

ロック

システム・テーブル・エントリ 47