



リファレンス・マニュアル：テーブル

Adaptive Server® Enterprise

15.7 ESD #2

ドキュメント ID : DC37455-01-1572-01

改訂 : 2012 年 7 月

Copyright © 2012 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。

Sybase の商標は、the Sybase trademarks page (<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。

このマニュアルに記載されている SAP、その他の SAP 製品、サービス、および関連するロゴは、ドイツおよびその他の国における SAP AG の商標または登録商標です。

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 Oracle およびその関連会社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

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	3
	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	5
	timestamp カラム	5
	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	6
	システム・テーブル使用上の規則	7
	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	8
	予約済みカラム	8
	システム・テーブルの更新	9
	システム・テーブル上のトリガ	9
	syblicenseslog	10
	sysalternates	11
	sysaltusages	12
	sysattributes	14
	sysauditoptions	17
	sysaudits_01 - sysaudits_08	18
	syscacheinfo	20
	syscachepoolinfo	21
	syscharsets	23
	syscolumns	24
	syscomments	27
	sysconfigures	29
	sysconstraints	31
	syscoordinations	32
	syscurconfigs	33
	sysdatabases	36
	sysdepends	40

sysdevices.....	41
sysencryptkeys.....	43
sysengines.....	45
sysgams.....	46
sysindexes.....	47
sysinstances.....	50
sysjars.....	51
syskeys.....	52
syslanguages.....	53
syslisteners.....	54
syslocks.....	55
sysloginroles.....	57
syslogins.....	58
syslogs.....	61
syslogshold.....	62
sysmessages.....	63
sysmonitors.....	64
sysobjects.....	65
sysoptions.....	70
syspartitionkeys.....	72
syspartitions.....	73
syspoolinfo.....	75
sysprocedures.....	76
sysprocesses.....	77
sysprotects.....	80
sysquerymetrics.....	83
sysqueryplans.....	85
sysreferences.....	86
sysremotelogins.....	87
sysresourcelimits.....	88
sysroles.....	89
syssecmechs.....	90
syssegments.....	91
sysservers.....	92
syssessions.....	94
syslices.....	95
sysssrroles.....	96
sysstatistics.....	97
sysabstats.....	98
systhresholds.....	100
sys timeranges.....	102
sys transactions.....	103
sys types.....	105
sysusages.....	108

	sysusermessages	109
	sysusers	110
	sysxtypes	111
第 2 章	dbccdb のテーブル.....	113
	dbccdb 作業領域	113
	dbccdb ログ	115
	dbcc_config	116
	dbcc_counters	117
	dbcc_exclusions	118
	dbcc_fault_params	119
	dbcc_faults	120
	dbcc_operation_log	121
	dbcc_operation_results	122
	dbcc_types	123
第 3 章	モニタリング・テーブル	131
	monCachedObject	132
	monCachePool.....	133
	monCachedProcedures	134
	monCachedStatement	135
	monCIPC.....	138
	monCIPCEndpoints.....	139
	monCIPCLinks	140
	monCIPCMesh	141
	monCLMObjectActivity	143
	monClusterCacheManager	145
	monCMSFailover	146
	monDataCache	147
	monDBRecovery	149
	monDBRecoveryLRTypes.....	151
	monDeadLock	152
	monDeviceIO	155
	monDeviceSpaceUsage.....	156
	monEngine	157
	monErrorLog	159
	monFailoverRecovery	160
	monInmemoryStorage.....	161
	monIOController.....	162
	monIOQueue.....	163
	monLicense.....	164
	monLocks.....	165
	monLockTimeout.....	167

monLogicalCluster	172
monLogicalClusterAction	174
monLogicalClusterInstance	175
monLogicalClusterRoute	176
monNetworkIO	177
monOpenDatabases	178
monOpenObjectActivity	180
monOpenPartitionActivity	185
monPCIBridge	190
monPCIEngine	191
monPCISlots	192
monPCM	193
monProcedureCache	195
monProcedureCacheMemoryUsage	196
monProcedureCacheModuleUsage	197
monProcess	198
monProcessActivity	200
monProcessLookup	202
monProcessMigration	203
monProcessNetIO	204
monProcessObject	205
monProcessProcedures	206
monProcessSQLText	208
monProcessStatement	209
monProcessWaits	210
monProcessWorkerThread	211
monRepLogActivity	212
monRepScanners	214
monRepScannersTotalTime	215
monRepSenders	216
monSpinlockActivity	217
monSQLRepActivity	218
monSQLRepMisses	219
monState	220
monStatementCache	221
monSysLoad	222
monSysPlanText	223
monSysSQLText	224
monSysStatement	225
monSysWaits	227
monSysWorkerThread	228
monTableColumns	229
monTableCompression	231
monTableParameters	232

	monTables	233
	monTableTransfer	234
	monTask	235
	monTempdbActivity	236
	monThread	237
	monThreadPool	238
	monWaitClassInfo	239
	monWaitEventInfo	240
	monWorkload	241
	monWorkloadPreview	242
	monWorkloadProfile	243
	monWorkloadRaw	244
	monWorkQueue	245
第 4 章	sybpcidb のテーブル	247
	pca_jre_arguments	248
	pca_jre_directives	249
	pci_arguments	250
	pci_directives	251
	pci_slotinfo	252
	pci_slot_syscalls	253
索引		255



システム・テーブル

システム・テーブルは、Sybase® から提供されます。Adaptive Server® 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	ページ
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	7

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。

システム・テーブル	内容
syscurconfigs	Adaptive Server が現在使用しているすべての設定パラメータに関する情報。
sysdatabases	Adaptive Server のデータベースごとに 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 (<code>us_english</code> を除く) ごとに 1 つのロー。
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。
sysresourcelimits	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。
syssservers	リモート Adaptive Server ごとに 1 つのロー。
syssessions	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のみ使用される。 <code>syssessions</code> 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。
sysssvroles	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` データベースだけにあります。

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。

システム・テーブル	内容
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つのローがあります。

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。

sysystemdb 内のシステム・テーブル

分散トランザクションのリモート・パーティシパントごとに1つのローで構成される、[syscoordinations](#) システム・テーブルは [sysystemdb](#) だけにあります。

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が含まれ、SQL 定義文を形成。
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 または image データのあるテーブルごとに1つのロー。
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。
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 には維持されない)。
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、およびテンポラリー・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。

システム・テーブル	内容
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。内部定義を形成。
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(ディスクの一部分の集まり)ごとに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つまたは複数のロー。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。
systabstats	テーブルごとに1つのロー、およびノンクラスタード・インデックスごとに1つのロー。
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。
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、デバッグ用に、使用されているシステムに [sybdiagdb](#) データベースを作成することができます。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](#)」(10 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 の停止に関連するライセンス情報について参照する必要がある場合があります。

Cluster Edition でのシステム・テーブルの使用

この項では Cluster Edition でのシステム・テーブルの一般的変更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 のノンクラスタ・エディションの identity カラムの動作とは異なります。Cluster Edition では、identity 値がユニークであることを保証しますが、パフォーマンス上の理由から、値は単調に増加しないことがあります。

ノンクラスタの 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 で記録サイズと記録動作に変更がないため、identity-burn-max は、ノンクラスタの 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・インスタンスからのみローを取得します。

- 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 入力可能なカラムは、“null” キーワードを使用して説明され、本書のテーブルのカラム説明にリストされます。

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、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できます。詳細については、『セキュリティ管理ガイド』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、次のように実行します。

```
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
- syssslices
- 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、システム・テーブルに通常必要な更新や追加を行うのに使用するべきシステム・プロシージャが含まれています。

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、『セキュリティ管理ガイド』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、Adaptive Server はエラー・メッセージを返し、トリガの作成を取り消す。

syblicenseslog

master データベースのみ

説明

syblicenseslog には、24 時間間隔で Adaptive Server で使用されるライセンスの最大数の更新ごとに、1 つのローが含まれています。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”でオブジェクト 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 = ライセンス数は制限を超過していない • 1 = ライセンス数は制限を超過している • -1 = ハウスキーピングはライセンス数をモニタできない
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、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・ダンプに保存されているもの
- 変更されたため、変更済みページ・セクションに再配置されているもの

『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	smallint	アーカイブ・データベースのデータベース ID。
location	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ます。
lstart	unsigned int	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。
size	unsigned int	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	int	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	int	ページの連続するブロックが存在するデバイス番号。
segmap	int	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す。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、サポートされるアーカイブ・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。スレッシュホールド・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のできるスクラッチ・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。

`sysaltusages` には、`dbid`、`location`、`lstart` のユニーク・クラスタード・インデックス (`csysaltusages`) が含まれます。

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、プロシージャなどのオブジェクトのプロパティを定義します。**sysattributes** は、(いろいろなシステム・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つのローを持っています。**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指定する。
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 (アプリケーション名など)。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 環境におけるオブジェクトに対する文字列の識別子 (アプリケーション名など)。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 など。オブジェクトがテーブルの一部 (インデックスなど) の場合、このカラムには関連テーブルのオブジェクト ID が含まれる。
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 を表します。 • 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> を指定する。
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。

名前	データ型	説明
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。
object_datetime	null	属性に対する datetime 値。使用方法は属性を使用するモジュールによって変わりますが、通常は属性が作成された日付と時刻を表します。

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 とともに使用します。

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
LR	ログイン・プロファイル
P	プロシージャ
T	表
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
SP	キー・ペア生成時間

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

sysbsecurity 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つのみです。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つのインストール環境では最大8つの監査テーブルを持つことができます。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
<code>event</code>	<code>smallint</code>	監査されるイベントのタイプ。
<code>eventmod</code>	<code>smallint</code>	イベントに関する詳細情報。値は次のとおり。 <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。
<code>spid</code>	<code>smallint</code> <code>int (Cluster Edition)</code>	監査レコードを書き込むプロセスのサーバ・プロセス ID。
<code>eventtime</code>	<code>datetime</code>	監査イベントの日付と時刻。
<code>sequence</code>	<code>smallint</code>	単一イベント内のレコードのシーケンス番号。一部のイベントは、複数の監査レコードを必要とする。
<code>suid</code>	<code>smallint</code>	監査イベントを実行したユーザのサーバ・ログイン ID。
<code>dbid</code>	<code>int null</code>	監査されるイベントが発生したデータベースの ID、または、オブジェクト、ストアド・プロシージャ、トリガが存在するデータベースの ID (イベントのタイプによる)。
<code>objid</code>	<code>int null</code>	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。
<code>xactid</code>	<code>binary(6) null</code>	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。
<code>loginname</code>	<code>varchar(30) null</code>	<code>suid</code> に対応するログイン名。
<code>dbname</code>	<code>varchar(30) null</code>	<code>dbid</code> に対応するデータベース名。
<code>objname</code>	<code>varchar(255) null</code>	<code>objid</code> に対応するオブジェクト名。
<code>objowner</code>	<code>varchar(30) null</code>	<code>objid</code> の所有者名。
<code>extrainfo</code>	<code>varchar(255) null</code>	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 を参照してください。
<code>nodeid</code>	<code>tinyint null</code>	今後のために予約済み (クラスタ環境では使用不可)。
<code>instanceid</code>	<code>tinyint</code>	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。

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、 alter table コマンドでは、 add column または drop constraint 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	set proxy 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。

syscacheinfo

master データベース

説明

データ・キャッシュに関する情報を提供する。

syscacheinfo はデータ・キャッシュおよびプールの設定に関する情報を提供する master データベースのビューです。

sa_role の役割を持つユーザのみがビューにアクセスできます。

カラム

syscacheinfo のカラムは次のとおりです。

名前	データ型	説明
cache_name	varchar(30)	このプールが割り当てられているキャッシュの名前。
cache_status	varchar(8)	キャッシュのステータス。次のいずれか。 <ul style="list-style-type: none"> • Active • Pend/Act • Act/Del
cache_type	varchar(16)	キャッシュのタイプ。次のいずれか。 <ul style="list-style-type: none"> • Mixed, HK Ignore • Mixed • Log Only • In-Memory Storage • Default
config_size	float	キャッシュ用として現在設定されているサイズ (MB)。run_size カラムに表示されるキャッシュの実サイズと値が異なることがある。
run_size	float	キャッシュに割り当てられている現在のメモリ容量 (MB)。config_size カラムに表示される設定されたサイズと値が異なることがある。
config_replacement	varchar(11)	現在設定されているバッファ置換方式。なし、または次のいずれか。 <ul style="list-style-type: none"> • Strict LRU • Relaxed LRU
run_replacement	varchar(11)	現在のキャッシュ用バッファ置換方式。なし、または次のいずれか。 <ul style="list-style-type: none"> • Strict LRU • Relaxed LRU
config_partitions	int	データ・キャッシュ用に設定されたパーティション数。
run_partitions	int	データ・キャッシュ用の現在のパーティション数。
overhead	numeric	データ・キャッシュ用のメモリ・オーバーヘッド容量。
cacheid	int	データ・キャッシュの ID。
instanceid	int	インスタンスの ID (ノンクラスタード・サーバでは、ゼロ)。
scope	varchar(6)	Cluster Edition でデータ・キャッシュがローカルかグローバルかを示す。値は、ノンクラスタード・サーバでは常にグローバル。

syscachepoolinfo

説明 データ・キャッシュの設定情報を含む各データ・キャッシュ・プールのローを表示する。このビューは `syscacheinfo` ビューと `syspoolinfo` ビューを統合したビューです。

カラム `sa_role` の役割を持つユーザのみがビューにアクセスできます。
`syscacheinfo` のカラムは次のとおりです。

名前	データ型	説明
<code>cache_name</code>	<code>varchar(30)</code>	このプールが割り当てられているキャッシュの名前。
<code>cache_status</code>	<code>varchar(8)</code>	キャッシュのステータス。次のいずれか。 <ul style="list-style-type: none"> • Active • Pend/Act • Act/Del
<code>cache_type</code>	<code>varchar(16)</code>	キャッシュのタイプ。次のいずれか。 <ul style="list-style-type: none"> • Mixed, HK Ignore • Mixed • Log Only • In-Memory Storage • Default
<code>cache_config_size</code>	<code>float</code>	キャッシュ用として現在設定されているサイズ (MB)。 <code>run_size</code> カラムに表示されるキャッシュの実サイズと値が異なることがある。
<code>cache_run_size</code>	<code>float</code>	キャッシュに割り当てられている現在のメモリ容量 (MB)。 <code>config_size</code> カラムに表示される設定されたサイズと値が異なることがある。
<code>cache_config_replacement</code>	<code>varchar(11)</code>	現在設定されているバッファ置換方式。なし、または次のいずれか。 <ul style="list-style-type: none"> • Strict LRU • Relaxed LRU
<code>cache_run_replacement</code>	<code>varchar(11)</code>	現在のキャッシュ用バッファ置換方式。なし、または次のいずれか。 <ul style="list-style-type: none"> • Strict LRU • Relaxed LRU
<code>cache_config_partitions</code>	<code>int</code>	データ・キャッシュ用に設定されたパーティション数。
<code>cache_run_partitions</code>	<code>int</code>	データ・キャッシュ用の現在のパーティション数。
<code>cache_overhead</code>	<code>numeric</code>	データ・キャッシュ用のメモリ・オーバーヘッド容量。
<code>pool_io_size</code>	<code>varchar(3)</code>	このプール用に I/O 実行に使用するバッファのサイズ (KB)。

名前	データ型	説明
pool_config_size	float	プールに割り当てられている現在のメモリ容量 (MB)。run_size カラムに表示される容量と値が異なることがある。
pool_run_size	float	プールに割り当てられている現在のメモリ容量 (MB)。
pool_apf_percent	int	プールのバッファのうち、非同期プリフェッチでキャッシュに読み込まれ、バッファを保持するために使用できるバッファの比率。
pool_wash_size	varchar(10)	プールのウォッシュ・エリアのサイズ (MB)。
cacheid	int	データ・キャッシュの ID。
instanceid	int	インスタンスの ID (ノンクラスタード・サーバでは、ゼロ)。
scope	varchar(6)	Cluster Edition でデータ・キャッシュがローカルかグローバルかを示す。値は、ノンクラスタード・サーバでは常にグローバル。

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 ~ 1999 の番号は文字セットを示す。2000 ~ 2999 の番号はソート順を示す。
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 の ID 番号は Sybase により予約されている。
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 の文字セット・ローは、このテーブルに存在していなければなりません。
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。7ビット ASCII 文字 A ~ Z、a ~ z、数字 0 ~ 9、およびアンダースコア () のみを使用でき、文字で始まる必要があります。
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 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 つのローが含まれています。

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 (非プロシージャ・エント리는 0)。
colid	smallint	カラム ID
status	tinyint	<ul style="list-style-type: none"> 0 ~ 2 ビット (値 1、2、4) は、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 = 常に複写 10 = 変更された場合のみ複写 00 = 決して複写しない 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) - 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 および 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。

名前	データ型	説明
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 <code>access_server</code> 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。
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。 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。
xbid	int null	クラスのデータベース ID。システム・クラスの場合は、値は -1。それ以外の場合は、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	<code>sysprocedures</code> 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、『セキュリティ管理ガイド』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。

名前	データ型	説明
status2	int null	<p>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。</p> <ul style="list-style-type: none"> 0x00000001、値 1 – パラメータ・モード “in” 0x00000002、値 2 – パラメータ・モード “out” <p>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。</p> <ul style="list-style-type: none"> 0x00000010、値 16 – カラムは計算カラム。 0x00000020、値 32 – カラムは実体化された計算カラム。 0x00000040、値 64 – カラムはビュー内の計算カラム。 0x00001000、値 4096 – 暗号化カラムには復号化デフォルトがある。 <p>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。</p> <ul style="list-style-type: none"> 0x80、値 128 – カラムは暗号化されている。 0x100、値 256 – カラムは初期化ベクトルによって暗号化されている。 0x200、値 512 – カラムはランダム埋め込みによって暗号化されている。 0x400、値 1024 – プロキシ・テーブルは暗号化されている。 0x1000、値 4096 – 暗号化カラムには復号化デフォルトがある。 0x20000、値 131072 – カラムは圧縮されないと明示的に定義されている。 0x00040000、値 262144 – ロー内に作成された LOB カラムのユーザ指定のロー内の長さまたは抽出されたロー内の長さ。
status3	int	0x0001、値 1 – 関数ベースのインデックス・キーの隠れた計算カラムを示す。
computedcol	int	計算カラム定義のオブジェクト ID を格納する。
encrytype	int null	暗号化された状態でのデータ型。
lobcomp_lvl	tinyint	ラージ・オブジェクトに定義されたカラムの圧縮レベル。
enclen	int null	暗号化データの長さ。
enckeyid	int null	キーのオブジェクト ID。
enckeydb	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。
enckdate	datetime null	暗号化キーの作成日。 <code>sysobjects.crdate</code> からコピーされる。
inrowlen	smallint	ロー内に作成された LOB カラムのユーザ指定のロー内の長さまたは抽出されたロー内の長さを格納する。

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 ローを使用できます。

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 (非プロシージャ・エントリは 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、およびプロシージャについてのシステム提供コメント。 1 - オブジェクトまたはカラムを説明するエントリの追加についてのユーザ提供コメント。
language	smallint	予約済み。
text	varchar(255) null	SQL 定義文の実際のテキスト。
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。1 以上になるのは、32,768 ローを超えるプロシージャ・テキストのあるプロシージャのみ。

名前	データ型	説明
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。 0x8 - SYSCOM_SHARED_INLINE_DEF はテキストが共有可能なインライン・デフォルトであることを示す。
version	smallint null	このロー用の隠しテキストの暗号化に使用されるアルゴリズムをエンコードする暗号化のバージョン。次のいずれか。 <ul style="list-style-type: none"> Null - 隠しテキストの暗号化なし 1 - (デフォルト) Adaptive Server 15.0 以降のバージョンで使用される Adaptive Server 難読化アルゴリズム 2 - (オプション) Advanced Encryption Standard (“AES”) 強力な暗号化
partitionid	int null	それ以外は null。
enckeyid	int null	version が 2 の場合は、Adaptive Server が現在のデータベース内の sysencryptkeys のキー・オブジェクトの隠しテキストを暗号化するのに使用した暗号化キー ID。それ以外は、Adaptive Server は enckeyid に null 値を使用する。

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。これらの文は Adaptive Server アップグレード・プロセスで必要です。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 を実行します。文が削除された場合、文を再作成するため、文を作成したオブジェクトを再作成するか、オブジェクトを作成したアプリケーションを再インストールする必要があります。

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 select on syscomments.text 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第 1 巻』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 データ型のパラメータのユーザ変更値。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	16 進	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	読み込み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

ステータス・タイプ	10 進	16 進	説明
CFG_EXTRAS_OPTION	512	0x200	CFG_EXTRAS 用のパラメータ (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

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 つのローも含まれています。

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。 <ul style="list-style-type: none"> • 1 – Begun • 4 – Prepared • 7 – Committed • 9 – In AbortTrans
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 によって調整される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・サーバでは null) 。
nodeid	tinyint null	クラスタ環境以外では使用不可 – 今後のために予約済み。
instanceid	tinyint	クラスタ環境のみ – インスタンスの ID

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 つのエントリが含まれています。

コラム syscurconfigs のコ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 データ型のパラメータの現在の実行値。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。表 1-4 を参照してください。
value2	varchar(255) null	character データ型のパラメータの現在の実行値を示す。integer 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、および 10。
datatype	int null	設定パラメータのデータ型。
message_num	int null	このパラメータ用の <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・プールの非同期プ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)。

名前	データ型	説明
instanceid	tinyint	<p>インスタンスの ID (クラスタ環境のみで使用可能)。</p> <ul style="list-style-type: none"> • Not applicable – パラメータにユニットなし • Megabytes • Memory pages (2K) • Number – 項目数 • Virtual pages (2K) • Clock ticks – クロック・チック数 • Logical pages • Microseconds • Percent • Milliseconds • Ratio • Seconds • Switch – ブール値 • Minutes • ID – ID 番号 • Hours • Name • Days • Rows • Bytes • Kilobytes
type	varchar(10) null	<p>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指定する。値は次のとおり。</p> <ul style="list-style-type: none"> • Dynamic – すぐに有効になる • Static – Adaptive Server の再起動後有効になる

表 1-4: status カラムの説明

ステータス・タイプ	10 進	16 進	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	読み込み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 用のパラメータ (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	16 進	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

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、sybssystemprocs 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、sybsecurity 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
name	sysname	データベースの名前。
dbid	smallint	データベース ID。
suid	int	データベース所有者のサーバ・ユーザ ID。
status	smallint	ユーザが sp_dboption で設定できる制御ビット。詳細については、 表 1-5 を参照。
version	smallint	未使用
logptr	int	トランザクション・ログへのポインタ。
crdate	datetime	作成した日付。
dumptrdate	datetime	最新の dump transaction の日付。
status2	smallint null	追加の制御ビット (表 1-6 (37 ページ) を参照)。
audflags	int null	データベースの監査設定。
deftabaud	int null	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。
defvwaud	int null	ビューのデフォルト監査設定を定義するビットマスク。
defpraud	int null	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。
def_remote_type	smallint null	ストアド・プロシージャ sp_addobjectdef 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。
def_remote_loc	varchar(349) null	ストアド・プロシージャ sp_addobjectdef 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。
status3	int null	追加の制御ビット。
status4	int null	追加の制御ビット。
audflags2	varbinary(16) null	今後のために予約済み。
instanceid	tinyint	インスタンスの ID (Cluster Edition のみ)。

名前	データ型	説明
durability	int	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
lobcomp_lvl	tinyint	LOB 圧縮レベル。

表 1-5 に、status カラムのビット表示をリストします。

表 1-5: sysdatabases テーブルのステータス制御ビット

10 進	16 進	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	select into/bulkcopy、ユーザによって設定可能。
8	0x08	trunc log on chkpt、ユーザによって設定可能。
16	0x10	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)。 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 で削除できる。
512	0x200	ddl in tran、ユーザによって設定可能。
1024	0x400	read only、ユーザによって設定可能。
2048	0x800	dbo use only、ユーザによって設定可能。
4096	0x1000	single user、ユーザによって設定可能。
8192	0x2000	allow nulls by default、ユーザによって設定可能。

表 1-6 に、status2 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 status2 制御ビット

10 進	16 進	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。

10 進	16 進	ステータス
128	0x0080	データベースに疑わしいページがある。
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 1-7 に、`status3` 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 `status3` 制御ビット

10 進	16 進	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。

表 1-8 に、`status4` 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 `status4` 制御ビット

10 進	16 進	ステータス
512	0x0200	インメモリ・データベースには、テンプレート・データベースがある。
4096	0x1000	データベースは、インメモリ・データベースである。

10 進	16 進	ステータス
16384	0x4000	64 このデータベースでは、ビットのアトミック演算が可能である。
16777216	0x01000000	データベースのすべてのテーブルはページ圧縮で作成される。
33554432	0x02000000	データベースのすべてのテーブルはロー圧縮で作成される。

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。
deppnumber	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 *</code> 文で使用される場合は <code>on</code> 。
resultobj	bit	オブジェクトが更新されている場合は <code>on</code> 。
readobj	bit	オブジェクトが読み込まれている場合は <code>on</code> 。
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、述部付きの権限など、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`、`deppnumber` カラムに、ユニーク・クラスター・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 番号を特定するスクリプトとストアド・プロシージャを変更する必要があります。

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ミング・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。
phname	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” はデフォルトでもある物理ディスクを示します。

表 1-9: status カラムのビット表示

10 進	16 進	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、dsync 設定を使用する (書き込みは物理メディアにフラッシュされる)。

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	16 進	ステータス
1	0x01	このデバイスではダイレクト I/O 有効化されています。

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。

フィールド	データ型	説明
<code>id</code>	<code>int</code>	暗号化キー ID。
<code>ekalgorithm</code>	<code>int</code>	暗号化アルゴリズム。
<code>type</code>	<code>smallint</code>	キーの種類。値は次のとおり。 <ul style="list-style-type: none"> • <code>0x1</code> (10 進 1) – 対称キー • <code>0x4</code> (10 進 4) – デフォルト・キー • <code>0x10</code> (10 進 16) – キー・コピー • <code>0x40</code> (10 進 64) – リカバリ・キー・コピー
<code>status</code>	<code>int</code>	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"> • <code>0x1</code> (10 進 1) – キーは初期化ベクトルを使用 • <code>0x2</code> (10 進 2) – キーはランダム埋め込みを使用 • <code>0x4</code> (10 進 4) – 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 • <code>0x8</code> (10 進 8) – ログイン・アクセスのために暗号化されたキー • <code>0x10</code> (10 進 16) – ログイン・パスワードを使用して暗号化されたキー・コピー • <code>0x20</code> (10 進 32) – システム暗号化パスワードを使用して暗号化されたキー・コピー • <code>0x100</code> (10 進 256) – ユーザ・パスワードを使用して暗号化されたキー
<code>eklen</code>	<code>smallint</code>	ユーザが指定したキーの長さ。
<code>value</code>	<code>varbinary(1282)</code>	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 によりシステム暗号化パスワード、ユーザ指定のパスワード、またはログイン・パスワードから抽出した 128 ビット・キーを使用する。
<code>uid</code>	<code>int null</code>	キー・コピーが割り当てられているユーザのユーザ ID。
<code>eksalt</code>	<code>varbinary(20)</code>	暗号化キーの復号化の検証に使用する乱数値。
<code>ekpairid</code>	<code>int null</code>	未使用。
<code>pwdate</code>	<code>datetime null</code>	最後にパスワードを変更した日付。
<code>expdate</code>	<code>int null</code>	未使用。
<code>ekpwdwarn</code>	<code>int null</code>	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	16 進	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"> プロセス・モード - オペレーティング・システムのプロセス ID スレッド・モード - オペレーティング・システムのスレッド (LWP) ID
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)	次のいずれか。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 プロセス数。
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 ページは、データベースのすべてのアロケーション・ユニットのビットマップを、アロケーション・ユニットごとに 1 ビットずつ格納します。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 カラムや 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属するテーブルの 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 のロー外構造 (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 ページあたりのローの最大数。
minlen	smallint	ローの最小サイズ。

名前	データ型	説明
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。
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	16 進	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。

10 進	16 進	ステータス
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。
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) インデックス。インデックスは別のソート順で作成された。

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 status ビット

10 進	16 進	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 カラムを含める。
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) が有効化されていない。
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化されていない。
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、新しいエクステンツにコピーする必要はない。
512	0x0200	インデックスは、DOL のクラスタード・インデックス。
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)。
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 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 つのローを持ちます。

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 名。
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： 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 でないキーの数。
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 の言語ごとに 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、いつでも Adaptive Server で利用することができます。

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” などの日付の順序。
datefirst	tinyint	週の最初の曜日。月曜日は 1、火曜日は 2、以下、水曜日～日曜日は 3～7。
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” などの公式言語名。
alias	varchar(30) null	“français” などの代替言語名。
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～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～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を問い合わせるとき、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 の名前と、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	unsigned 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	ロックが属するファミリー (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)。 <code>fid</code> の値は次のとおり。
		<ul style="list-style-type: none"> 0 — <code>spid</code> 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 (<code>spid</code>) は、並列で文を実行しているファミリーのメンバである。 <p>この値が <code>spid</code> と同じ場合、そのタスクがファミリー内でクエリを並列実行しているコーディネーティング・プロセスであることを示す。</p>
int (Cluster Edition)		
context	tinyint	ロック要求のコンテキスト・タイプ。 <code>context</code> の値については、 表 1-15 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)。

[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 テーブルの type 制御ビット

10 進	16 進	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、context カラムの値をリストします。

表 1-15: 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 並列実行されているクエリ。
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、役割を持つサーバ・ログインまたはログイン・プロファイル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`、`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`) に対応する `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 またはログイン・プロファイル ID
<code>srid</code>	<code>int</code>	サーバの役割 ID。次のいずれか 1 つ。 <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>webservices_role</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 = デフォルトはオフ • 1 = デフォルトはオン

注意 `alter login` 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・アカウントまたはログイン・プロファイルごとに 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 またはログイン・プロファイル ID。
status	smallint	アカウントのステータス (表 1-16 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。ログイン・プロファイルがログイン・アカウントに関連付けられている場合、カラムはログイン・ローには適用されない。
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。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。ログイン・プロファイルがログイン・アカウントに関連付けられている場合、このカラムはログイン・ローには適用されない。
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。次のいずれか。 <ul style="list-style-type: none"> 0x00000001 – ユーザ作成のテーブルの参照に成功 0x00000002 – 失敗 0x00000004 – ユーザ作成のビューの参照に成功 0x00000008 – 失敗 0x00000010 – ユーザ cmdtext 監査 0x00000020 – 埋め込み必須 0x00000040 – ユーザ・アクションの監査はすべて成功 0x00000080 – ユーザ・アクションの監査はすべて失敗
fullname	varchar(30) null	ユーザのフルネーム。
srvname	varchar(30) null	AUTOCONNECT フラグがオンの場合、パスマスルー接続を確立する宛先サーバ名。

名前	データ型	説明
logincount	smallint null	ログインを失敗した回数。成功した場合に 0 にリセットされる。
procid	int null	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。ログイン・プロファイルがログイン・アカウントに関連付けられている場合、このカラムはログイン・ローには適用されない。
lastlogindate	datetime	ユーザの前回ログイン時のタイムスタンプ。
crdate	datetime	ログインまたはログイン・プロファイル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)。
lockreason	int	ロックの理由。次のいずれか。 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。 4 - 非アクティブであったため、locksuid によってアカウントが自動的にロックされた。
lockdate	datetime	指定する値は次のとおりです。 <ul style="list-style-type: none"> • ログイン・アカウントがロックされている - syslogins.lockdate は、ログインがロックされたタイムスタンプを指定する。 • ログイン・アカウントはロックされていない、および <ul style="list-style-type: none"> • syslogins.lockdate が NULL ではない - ログインのロックが解除されたときのタイムスタンプを指定する。 • syslogins.lockdate が NULL - ログインが一度もロックされていないことを指定する。
crsuid	int	ログインまたはログイン・プロファイルの作成者のサーバ・ユーザ ID。
lpid	int	ログイン・プロファイル ID。 <ul style="list-style-type: none"> • null - ログイン・アカウントはデフォルトのログイン・プロファイルに関連付けられている (ある場合)。 • -1 - ログイン・プロファイルは、ログイン・アカウントで無視される。 • suid - ログイン・プロファイル ID。

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” 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 1-16: syslogins テーブルのステータス制御ビット

10 進	16 進	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	<code>exempt inactive lock</code> の値が <code>TRUE</code> に設定されていることを示す。ログイン・プロファイル・ローには適用されない。
16	0x10	<code>OMNI</code> : 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。
512	0x200	ログイン・プロファイルを示す。
1536	0x200 0x400	デフォルトのログイン・プロファイルを示す。
2048	0x800	ログイン・プロファイルで指定された認証メカニズムを示す。

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。

syslogs

すべてのデータベース

説明

`syslogs` には、トランザクション・ログが格納されます。Adaptive Server がリカバリおよびロールフォワードで使用します。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取るためのローが `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
<code>xactid</code>	<code>binary(6)</code>	トランザクション ID。
<code>op</code>	<code>tinyint</code>	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	unsigned int	最も古いトランザクションで定義された <code>syslogs</code> のアクティブ部分の開始ページ番号 (Replication Server の場合は <code>syslogs</code> 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 (ある場合) の ID。それ以外の場合は 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刻 (Replication Server のトランケーション・ポイントが設定されたとき)。
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、 <code>begin transaction</code> で定義される名前。ただし、 <code>begin transaction</code> 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 (\$) で始まる名前を持ち、オペレーションに対する名前が付けられる。または、Replication Server の “\$replication_truncation_point” と名前が付けられる。
xloid	int null	ロック所有権 ID。所有者がタスクの場合は <code>spid</code> に基づく ID、所有者がトランザクションの場合は <code>xdes</code> 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 はユーザの画面にエラー説明を表示します。

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。

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

sysencryptkeys の encrkeyid 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 • EK — 暗号化キー • F — SQLJ 関数 • 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 • PP — 述部権限 • 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 • RS — 事前計算済み結果セット • 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

名前	データ型	説明
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、schemacnt を更新する。
schemacnt	smallint	オブジェクトのスキーマの変更数 (ルールまたはデフォルトが追加されると増加する)。
sysstat2	int	追加の内部ステータス情報 (表 1-18 を参照)。
sysstat3	unsigned smallint	追加の内部ステータス情報 (表 1-19 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。エントリがトリガの場合は、テーブル ID。
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。
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。
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。
versionts	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 カラムの消去済みの最大値。 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。
spacestate	smallint null	内部使用のみ。
erlchgts	binary(8) null	内部使用のみ。
lobcomp_lvl	tinyint	LOB 圧縮レベル。

表 1-17 に、sysstat カラムのビット表示をリストします。

表 1-17: sysobjects テーブルの sysstat 制御ビット

10 進	16 進	説明
0	0x0	すべての無効なオブジェクト。
1	0x1	システム・オブジェクト。

10 進	16 進	説明
2	0x2	ビュー。
3	0x3	ユーザ・オブジェクト。
4	0x4	ストアド・プロシージャ。
5	0x5	述部付きの権限。
6	0x6	デフォルト値の仕様。
7	0x7	ドメイン・ルール。
8	0x8	トリガ・プロシージャ。
9	0x9	参照整合性制約。
10	0xA	SQL 関数
11	0xB	拡張タイプ。
12	0xC	ストアド・ファンクション。
13	0xD	計算カラム。
14	0xE	パーティション条件
15	0xF	暗号化キー
16	0x10	クラスタード・インデックスがある。
32	0x20	ノンクラスタード・インデックスがある。
64	0x40	オブジェクトがテーブルの場合、そのオブジェクトの変更が記録される。オブジェクトがプロシージャの場合、複写によってそのプロシージャの実行をサブスクライブできることを示す。
128	0x80	オブジェクトを作成中。
256	0x100	オブジェクトに疑わしいインデックスがあるため、 <code>dbcc reindex</code> を実行するまで読み込み専用としてのみ使用可能。
512	0x200	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。 <code>dbcc</code> を実行する必要がある。 <code>opentable</code> によって検査済み。
1024	0x400	オブジェクトは「偽」。つまり、オブジェクトは <code>tempdb</code> 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。
2048	0x800	オブジェクトは、クエリのコンパイルのために作成された定義時オブジェクトである。
4096	0x1000	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する。
8192	0x2000	<code>text</code> または <code>image</code> フィールドを含むオブジェクト。
16384	0x4000	未使用。
32768	0x8000	テーブルまたはプロシージャは複写される。

表 1-18 に、`sysstat2` 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 `sysstat2` 制御ビット

10 進	16 進	ステータス
0	0x00	非連鎖トランザクション・モード。
1	0x1	テーブルに参照制約がある。

10 進	16 進	ステータス
2	0x2	テーブルに外部キー制約がある。
4	0x4	テーブルに複数の検査制約がある。
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	オブジェクトは仮想ハッシュ・テーブルである。
256	0x100	ストアド・プロシージャ内の実行即時呼び出しで暗黙的な付与を許可する。
512	0x200	テーブルに可変長カラムがない。
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	オブジェクトは、所有者名によって複写される。
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 以降で作成された。
131072	0x20000	テーブルにクラスタド・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	混合型テーブル。
16777216	0x1000000	アクセス・ルール。
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のインデックスが含まれる。
2147483648	0x80000000	オブジェクトに拡張インデックスがある。

表 1-19 に、sysstat3 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 sysstat3 制御ビット

10 進	16 進	ステータス
256	0x0100	execute as owner 句で作成されたストアド・プロシージャ。
512	0x0200	execute as caller 句で作成されたストアド・プロシージャ。
2048	0x0800	テーブルに LOB 圧縮データがある。
4096	0x1000	テーブルはロー・レベル圧縮を使用する。

10 進	16 進	ステータス
8192	0x2000	テーブルはページ・レベル圧縮を使用する。
16384	0x4000	テーブルに圧縮データがある。
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 が含まれる。
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 time オプション • Bit 2 – スタートアップ・プロシージャ固有のオプション • Bit 3 – binary オプション
number	int		整数としてのスイッチ ID。

sysoptions は次のことを示します。

- runserver ファイルに -T オプションが設定されたトレース・フラグ。
- dbcc traceon(flag_number) または set switch serverwide on が設定されたトレース・フラグ。
- 特定のシステム・プロセス ID (SPID) に対して set switch on が設定されたトレース・フラグおよびスイッチ。

sysoptions には、sysoptions テーブルのクエリを実行するユーザーが参照可能なスイッチのみが表示されます。つまり、ユーザーは他の SPID によって set switch on でプライベートに設定されているスイッチを表示できません。ただし、runserver ファイルの -T オプション、dbcc traceon または set switch serverwide on を使用して有効にされたトレースフラグはすべてのユーザーが見ることができます。

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 (分割されていない) の各テーブル。
- 単一パーティション (分割されていない) 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
name	varchar(255)	パーティション名。
indid	smallint	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
id	int	テーブル ID。

名前	データ型	説明
partitionid	int	データまたはインデックス・パーティションの ID。
segment	smallint	パーティションがあるセグメントの ID。
status	int	内部ステータス情報。
dataoampage	unsigned int	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。
indoampage	unsigned 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	unsigned 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	unsigned 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 = テーブル全体をカバーするグローバル・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カバーするデータ・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。
cdataptname	varchar(255) null	データ・パーティションの名前。
lobcomp_lvl	tinyint	LOB 圧縮レベル。
ptndcompver	tinyint	使用したデータ圧縮アルゴリズムのバージョン。

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。

syspoolinfo

master データベース

説明 データ・キャッシュとプールに関する情報を提供します。
sa_role の役割を持つユーザのみがビューにアクセスできます。

カラム syspoolinfo のカラムは次のとおりです。

名前	データ型	説明
cache_name	varchar(30)	このプールが割り当てられているキャッシュの名前。
io_size	varchar(3)	このプール用に I/O 実行に使用するバッファのサイズ (KB)。
config_size	float	プールに割り当てられている現在のメモリ容量 (MB)。run_size カラムに表示される容量と値が異なることがある。
run_size	float	プールに割り当てられている現在のメモリ容量 (MB)。
apf_percent	int	プールのバッファのうち、非同期プリフェッチでキャッシュに読み込まれ、バッファを保持するために使用できるバッファの比率。
wash_size	varchar(10)	プールのウォッシュ・エリアのサイズ (MB)。
cacheid	int	データ・キャッシュの ID。
instanceid	int	インスタンスの ID (ノンクラスタード・サーバでは、ゼロ)。
scope	varchar(6)	Cluster Edition でデータ・キャッシュがローカルかグローバルかを示す。値は、ノンクラスタード・サーバでは常にグローバル。

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 エントリに収まらない場合、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 1-20 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 (非プロシージャ・エントリは 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 1-20 に、type カラムのビット表示をリストします。

表 1-20: sysprocedures テーブルの type 制御ビット

10 進	16 進	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	プロセス ID。
	int (Cluster Edition)	
kpids	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 1-21 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。
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	プロセスの累積 CPU 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	ブロックしているプロセスがある場合のプロセス ID。
	int (Cluster Edition)	
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	ワーカー・プロセスの親プロセスのプロセス ID。
	int (Cluster Edition)	
execlass	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 set proxy または set session authorization を実行している origsuid の suid 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機知のユーザの名前。 注意 Adaptive Server は 1 つ以上のスペースを clientname、clienthostname、clientapplname 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識されているホストの名前。
clientapplname	varchar(30) null	オプション - 現在のセッションで認識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 に接続したときの日時を示す。詳細については、『セキュリティ管理ガイド』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、『セキュリティ管理ガイド』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	(Cluster Edition) 位置合わせのために追加されるカラム。
lcid	int	(Cluster Edition) 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 1-21 に、status カラムの値をリストします。

表 1-21: sysprocesses の status カラムの値

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 <code>waitfor delay</code> 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・プロセスなどのプロセス。
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。
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 またはその他のリソース待ち (通常、プロセスは実行中であるが非常に大量のディスク I/O が行われていることを示す)。
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
id	int	パーミッションが適用されるオブジェクトの ID。扶養されているパーミッションが <code>create table</code> や <code>create default</code> などの場合は、ID は 0。
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	詳細については、表 1-22 を参照してください。
protecttype	tinyint	次の値のいずれか。 0 = 0 = grant with grant 1 = grant 2 = revoke
columns	varbinary(133)	この <code>select</code> 、 <code>update</code> 、 <code>decrypt</code> 、または <code>references</code> 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。 0 = すべてのカラムを示す。 1 = そのカラムにパーミッションが適用されることを示す。 NULL = 情報がないことを示す。 <code>columns</code> は、 <code>set session authorization</code> 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。
predid	int	述部付き権限のオブジェクト ID。
status	smallint	0x0001 - 権限 (または拒否) が含まれることを示す。

表 1-22: sysprotects action カラムの値

1 = alter any object owner *	67 = drop any default *
2 = alter any table *	68 = drop any function *
3 = change password *	70 = drop any object *
4 = checkpoint any database *	71 = drop any procedure *
5 = select builtin	72 = drop any rule *
6 = checkpoint *	73 = drop any table *
7 = create any default *	74 = drop any trigger *
8 = create any function *	75 = drop any view *
9 = create any index *	76 = dump database *
10 = create any object *	77 = dump any database *
11 = create any procedure *	79 = execute any function *
12 = create any rule *	80 = execute any procedure *
13 = create any table *	80 = identity_insert any table *
14 = create any trigger *	82 = identity_update any table *
15 = create any view *	81 = identity_insert *
16 = allow exceptional login *	82 = identity_update *
17 = dbcc checkalloc	85 = insert any table *
18 = dbcc checkalloc any database	86 = kill *
19 = map external file *	87 = kill any process *
20 = manage dump configuration *	88 = load database *
21 = dbcc checkcatalog	89 = load any database *
22 = dbcc checkcatalog any database	90 = manage service key *
25 = dbcc checkdb	91 = manage abstract plans *
26 = dbcc checkdb any database	92 = manage any encryption key *
29 = dbcc checkindex	93 = manage any esp *
30 = dbcc checkindex any database	94 = manage any execution class *
33 = dbcc checkstorage	95 = manage any login *
34 = dbcc checkstorage any database	96 = manage any login profile *
37 = dbcc checktable	97 = manage any object permission *
38 = dbcc checktable any database	98 = manage any remote login *
41 = dbcc checkverify	99 = manage any statistics *
42 = dbcc checkverify any database	100 = manage any user *
45 = dbcc fix_text	101 = manage auditing *
46 = dbcc fix_text any database	102 = manage checkstorage *
49 = dbcc indexalloc	103 = manage cluster *
50 = dbcc indexalloc any database	104 = manage data cache *
53 = dbcc reindex	105 = manage database *
54 = dbcc reindex any database	106 = manage database permissions *
57 = dbcc tablealloc	107 = manage disk *
58 = dbcc tablealloc any database	108 = manage lock promotion threshold *
61 = dbcc textalloc	109 = manage master key *
62 = dbcc textalloc any database	110 = manage replication *
65 = dbcc tune	111 = manage resource limit *
66 = delete any table *	112 = manage roles *

113 = manage security configuration *	149 = set switch *
114 = manage security permissions *	150 = show switch *
115 = manage server *	151 = references
116 = manage server configuration *	152 = truncate any audit table *
117 = manage server permissions *	153 = decrypt any table *
118 = monitor qp performance *	155 = manage column encryption key *
119 = monitor server replication *	156 = manage any database *
120 = mount any database *	167 = set proxy
121 = online any database *	193 = select
122 = online database *	195 = insert
123 = own any database *	196 = delete
125 = own database *	197 = update
126 = quiesce any database *	198 = create table
129 = references any table *	203 = create database
130 = report checkstorage *	207 = create view
131 = reorg any table *	221 = create trigger
132 = select any audit table *	222 = create procedure
133 = select any system catalog *	224 = execute
134 = select any table *	233 = create default
135 = set tracing any process *	235 = dump transaction
136 = setuser	236 = create rule
137 = shutdown *	253 = connect
138 = transfer any table *	280 = create function
139 = manage any thread pool *	282 = delete statistics
140 = truncate any table *	320 = truncate table
141 = unmount any database *	326 = update statistics
144 = update any security catalog *	347 = 347 = set tracing
145 = update any table *	353 = decrypt
146 = use any database *	354 = create encryption key
148 = use database *	368 = transfer table

注意 アスタリスク (*) が付いている action カラムのパーミッションは、細密なパーミッションが有効な場合に限り有効です。

インデックス

- id、action、grantor、uid、protecttype、predid 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 テキストに複数のローが必要になる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	リソース制限の超過により Resource Governor によってクエリがアボートされた回数。
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。

```
selet * 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、個々の概略クエリ・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。または A100 = テキスト・カラムに抽象プラン・テキストが含まれる。
sequence	smallint	SQL クエリまたは抽象プランのテキストに複数の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。
frgnbid	smallint null	参照している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・キーのあるテーブル) 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	1 番目 ~ 16 番目の参照元カラムのカラム ID。
refkey1 ~ refkey16	smallint	1 番目 ~ 16 番目の参照先カラムのカラム ID。
frgnbname	varchar(30) null	参照元テーブル (外部キーのあるテーブル) を含むデータベースの名前。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・キーのあるテーブル) を含むデータベースの名前。参照先テーブルが現在のデータベースにある場合は NULL。

sysreferences テーブルのステータス・ビットは次のとおりです。

10 進	16 進	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、frgnbname 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、frgnbname 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が使用できるサーバ・リソースの最大量を指定します。

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	systimeranges 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 = 両方
action	tinyint	違反したときの動作。 <ul style="list-style-type: none"> • 1 = 警告の発行 • 2 = クエリ・バッチのアボート • 3 = トランザクションのアボート • 4 = セッションの強制終了
limitvalue	int	制限の値。
scope	tinyint	ユーザ制限の範囲 (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”。
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”。

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。

syservers

master データベースのみ

説明 syservers 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 ServerTM、または Open ServerTM ごとに 1 つのローが含まれています。

カラム syservers 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 1-23 を参照)。
srvstatus2	unsigned int	オプションのビットマップ (表 1-24 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 1-25 を参照)。
srvsecmech	varchar(30) null	セキュリティ・メカニズム。
svrcost	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 (ミリ秒単位) を提供する。プロキシ・テーブルへのアクセス時に、Adaptive Server クエリ・オブティマイザがクエリのコストの評価のためのみ使用する。デフォルトは 1,000 ミリ秒。
srvprincipal	varchar(255) null	リモート・サーバの Kerberos プリンシパル名を指定する。デフォルト値は NULL。

表 1-23 に、srvstatus カラムのビット表示をリストします。

表 1-23: `srvstatus` カラムのステータス制御ビット

10 進	16 進	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。
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 1-24: `syssservers` テーブルの `srvstatus2` 制御ビット

10 進	16 進	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 1-25 に、`srvclass` カラムのサーバ・カテゴリをリストします。

表 1-25: サーバ・カテゴリは `syssservers` テーブル

<code>srvclass</code>	サーバ・カテゴリ
0	ローカル・サーバ
1	<code>sql_server</code> クラス・サーバ
3	<code>direct_connect</code> クラス・サーバ
4	DB2 クラス・サーバ
6	<code>sds</code> 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

- `srvid` 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。

syssslices

すべてのデータベース

説明

syssslices 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。syssslices 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 syssslices 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

syssslices 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 1-26 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 にリセットされる。
locksuid	int null	役割のロックを実行した人物。
lockreason	int null	役割がロックされた理由。
lockdate	datetime null	役割がロックされた日付と時刻。

表 1-26 に、status カラムのビット表示をリストします。

表 1-26: sysssrvroles テーブルのステータス制御ビット

10 進	16 進	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。

systabstats

すべてのデータベース

説明 `systabstats` 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 `systabstats` 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><code>systabstats</code> には、<code>text</code> または <code>image</code> オブジェクト (255) に関する統計は保持されていない。</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	unsigned 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	unsigned 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 回数。
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 回数。
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 回数。
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。
datarowsize	float	データ・ローの平均サイズ。
leafrowsize	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 1-27 を参照)。
plljoindegree	int	ネスト・ループ・ジョイン操作で使用される並列度。plljoindegree 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 (<code>systabstats</code> がこのフィールドを持つテーブル) のスキャン並列度。

名前	データ型	説明
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。
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) のテーブルまたはインデックスで可能な最大並列度。値 0 (ゼロ) は最大値が存在しないことを示す。クエリ・プロセッサは最大並列度を設定する。
emptypgcnt	unsigned 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。
partitionid	int	パーティション ID。
warmcachepgcnt	unsigned int	
statmoddate	datetime	ローが最後にディスクにフラッシュされた時刻。
unusedpgcnt	unsigned int	未使用のページ数。
oampagecnt	unsigned 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。
pioclmdata	real	
pioclmindex	real	
piocsmdata	real	
piocsminindex	real	
spare2	float	予約済み。
spare4	float	予約済み。
spare5	int	配置を揃えるための予備のフィールド。

systabstats テーブルのステータス・ビットは次のとおりです。

表 1-27: systabstats テーブルの status カラムのステータス・ビット

10 進	16 進	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	unsigned 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ビット 1 は 1、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 からアクティブな役割を決定 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	16 進	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

systimeranges

master データベースのみ

説明 systimeranges 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 systimeranges 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” の制限を表す。
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ています。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。有効な値： 0 - Resident Tx 1 - Failed-over Tx 2 - Tx by Failover-Conn
type	int	トランザクションのタイプを示す値。有効な値： 1 - Local 3 - External 98 - Remote 99 - Dtx_State
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。有効な値： 0 - None 1 - Syb2PC 2 - ASTC 3 - XA 4 - DTC
state	int	トランザクションの現在のステータスを示す値 (表 1-28 を参照)。
connection	int	接続ステータスを示す値。connection の値とステータスは次のとおり。 1 - Attached 2 - Detached
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	サーバ・プロセス ID、またはプロセスが分離されている場合は 0。
	int (Cluster Edition)	
masterbid	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。

名前	データ型	説明
xactname	varchar(255) null	トランザクション名または <i>XID</i> 。
srvname	varchar(30) null	リモート・サーバの名前 (ローカル・サーバでは 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 1-28: systransactions state カラム値

state 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

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、それ以外の場合は 0。
allownulls	bit	データ型に null が許可されているかどうかを示す。
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	カラムに IDENTITY プロパティがある場合は 1、ない場合は 0。
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	sysprocedures におけるアクセス・ルールのオブジェクト ID。

表 1-29 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 1-29: データ型名、階層、タイプ、ユーザ・タイプ

名前	階層	データ型	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

名前	階層	データ型	ユーザ・タイプ
image	32	34	20
date	33	49	37
time	34	51	38
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「ディスク割り当ての集まり」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 第 2 巻』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 以降では、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 番号を特定するスクリプトとストアド・プロシージャを変更する必要がある場合があります。

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	unsigned int	最初のデータベース (論理) ページ番号。
size	unsigned int	連続したデータベース (論理) ページ数。
vstart	int	仮想ページの開始番号。
unreservedpgs	unsigned int	割り当てられたエクステンツ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 以上である必要がある。
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 の場合、null。
dlevel	smallint null	with_log ビットを格納する。これは、メッセージをログに記録するための適切なルーチン呼び出すのに使用する。

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。このデータベース内でユニークであり、パーミッションの付与や取り消しに使用する。ユーザ ID 1 は “dbo”。
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) に設定されます。
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 に対する “allow duplicate rows” を持つユニークでないクラスタード・インデックス
- 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 クラスでは、クラス・ファイルが含まれる。データは、最大 255 バイトまでの長さがロー内に保存される。
xtbinaryoffrow	image	拡張型のオブジェクト・コード。Java クラスでは、クラス・ファイルが含まれる。データは、イメージ・カラムとして、ロー外に格納される。

- 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。

dbccdb のテーブル

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の出力を定義する 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	ページ
dbccdb 作業領域	113
dbccdb ログ	115

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) カラムからなります。

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

Description

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するものです。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 ではない場合にのみ、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 ではない場合にのみ、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。

See also

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

Description 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

Description 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です。 <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 」(123 ページ)を参照してください。
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 」(123 ページ)を参照してください。

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

dbcc_fault_params

Description 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。

各列 “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

Description `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、 <code>sysindexes</code> と <code>sysobjects</code> から抽出される。
indid	smallint	インデックスを示す。この値は、 <code>sysindexes</code> から抽出される。
partitionid	int	パーティションを示す。この値は、 <code>sysindexes</code> と <code>syspartitions</code> 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、 <code>sysdevices</code> から抽出される。
opid	smallint	実行された <code>dbcc</code> 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 (123 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：</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

Description 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

Description 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 (123 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 砺 “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

Description

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 スタアド・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、フォーマットの分類、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タアド・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) の数を指定する。

type_code	type_name	説明
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) の数を指定する。
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指定する。
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 カラムの数を指定する。
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ツ数を示す。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ツの数を示す。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。

type_code	type_name	説明
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。
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、データの無いページ数。
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ランケートされる。

type_code	type_name	説明
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。
10009	object id expected	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。 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。
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。

type_code	type_name	説明
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。
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 dbid 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。

type_code	type_name	説明
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、 page object id error と異なる。
100007	extent id error	dbcc checkstorage 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。 dbcc checkstorage によってこのエラーがレポートされる。 dbcc checktable 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 text データまたは image 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 text カラムまたは image 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、 wrong type error 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。
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。

type_code	type_name	説明
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・スレッシュ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。

type_code	type_name	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 textalloc 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	エクステンツ 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 2³¹ 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ります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。

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			(クラスタ環境のみ) 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 <code>syspartitions</code> 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)。
Processes Accessing	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。

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		(クラスタ環境のみ) 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。
LogicalReads	int	カウンタ	プールから読み込まれたバッファの数。
PhysicalWrites	int	カウンタ	このプールのデータに実行する書き込み操作の回数 (1 回の書き込み操作に複数ページが含まれる場合がある)。
APFReads	int	カウンタ	このプールにページをロードした非同期プリフェッチ (APF) 読み込み操作の数。
APFPercentage	int		このプールに設定された非同期プリフェッチ制限。
WashSize	int		メモリ・プールのウォッシュ・サイズ (キロバイト単位)。

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。

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 **statement statistics active** の各設定パラメータを有効にしてください。

コラム monCacheProcedures のコ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		(クラスタ環境のみ) 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)。
ExecutionCount	int	カウンタ	ストアド・プロシージャ・プランまたはツリーがキャッシュされてから Adaptive Server により実行された回数。
CPUTime	int	カウンタ	費やされた合計 CPU 時間 (ミリ秒単位)。
ExecutionTime	int	カウンタ	Adaptive Server でストアド・プロシージャ・プランまたはツリーの実行に費やした合計経過時間 (ミリ秒単位)。
PhysicalReads	int	カウンタ	実行した物理読み込み回数。
LogicalReads	int	カウンタ	読み込まれたページ数。
PhysicalWrites	int	カウンタ	実行した物理書き込み回数。
PagesWritten	int	カウンタ	書き込まれたページ数。

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。

注意 クロック周波数が異なる複数の CPU を使用するマシンでは、不正確な経過時間が報告される場合があります。

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。

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 cache size、および enable stmt cache monitoring の各設定パラメータを有効にしてください。

カラム

monCacheStatement のカラムは次のとおりです。

名前	データ型	説明
InstanceID	tinyint	(クラスタ環境のみ) 共有ディスク・クラスタ内のインスタンスの ID。
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 を使用します。
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。
StmtType	tinyint	
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	smallint	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。
UseCount	int	キャッシュ後の文へのアクセス回数。
StatementSize	int	キャッシュされた文のサイズ(バイト単位)。
MinPlanSizeKB	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)。

名前	データ型	説明
MaxPlanSizeKB	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)。
CurrentUsageCount	int	キャッシュされた文の同時ユーザの数。属性はカウンタ。
MaxUsageCount	int	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。属性はカウンタ。
NumRecompilesSchemaChanges	int	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。 update statistics 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。 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。属性はカウンタ。
NumRecompilesPlanFlushes	int	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。属性はカウンタ。
HasAutoParams	tinyint	パラメータ化されたリ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”。
ParallelDegree	tinyint	この文用に格納されているクエリが使用する並列度。
QuotedIdentifier	tinyint	set quoted_identifier 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	tinyint	文をコンパイルしたトランザクション独立性レベル。
TransactionMode	tinyint	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	tinyint	sa_role 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。
SystemCatalogUpdate	tinyint	プランをコンパイルしたときに allow catalog updates が有効な状態であったかどうかを示す。
MetricsCount	int	この文に対して測定基準が集計された回数。
MinPIO	int	この文の実行中に発生する物理 I/O の最小数。
MaxPIO	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。
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。
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。
AvgLIO	int	この文の実行中に発生する論理 I/O の平均数。
MinCpuTime	int	この文の実行で消費された最小 CPU 時間(ミリ秒単位)。
MaxCpuTime	int	この文の実行で消費された最大 CPU 時間(ミリ秒単位)。
AvgCpuTime	int	この文の実行で消費された平均 CPU 時間(ミリ秒単位)。
MinElapsedTime	int	この文に対する最小の経過実行時間。
MaxElapsedTime	int	この文に対する最大の経過実行時間。

名前	データ型	説明
AvgElapsedTime	int	この文に対する平均経過実行時間。
AvgScanRows	int	実行 1 回あたりに読み取られたスキャン済みローの平均数。
MaxScanRows	int	実行 1 回あたりに読み取られたスキャン済みローの最大数。
AvgQualifyingReadRows	int	read コマンドの実行 1 回あたりに該当したデータ・ローの平均数。
MaxQualifyingReadRows	int	クエリの実行 1 回あたりに該当したデータ・ローの最大数。
AvgQualifyingWriteRows	int	クエリの実行 1 回あたりに該当したデータ・ローの平均数。
MaxQualifyingWriteRows	int	クエリの実行 1 回あたりに該当したデータ・ローの最大数。
LockWaits	int	ロック待機の合計回数。
LockWaitTime	int	ロック待機に費やされた合計時間 (ミリ秒単位)。
SortCount	int	ソート・オペレーションの合計回数。
SortSpilledCount	int	ディスクにあふれたソート・オペレーションの合計回数。
TotalSortTime	int	ソートに費やされた合計時間 (ミリ秒単位)。
MaxSortTime	int	ソートに費やされた最大時間 (ミリ秒単位)。
DBName	varchar(30)	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。属性は null。
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。
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。
OptimizationGoal	varchar(30)	この文の最適化で使用する最適化目標。
OptimizerLevel	varchar(30)	この文の最適化で使用するオプティマイザ・レベル。
AdjustToParallel	int	利用可能なワーカー・スレッド数が不足し、クエリ・プランが要求する並列度でクエリを実行できない場合に、クエリが一部並列処理で実行されたことを示す。
AdjustToSerial	int	利用可能なワーカー・スレッド数が不足し、クエリを並行処理で実行できない場合に、クエリがシリアル処理で実行されたことを示す。
ThreadDeficit	int	ステートメント・キャッシュに追加されたので、ワーカー・スレッドの累積的な総数をこのクエリを実行するために使用できないことを示す。

monCIPC

説明

(クラスタ環境のみ) 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。

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	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リ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リ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

(クラスタ環境のみ) 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。

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	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 (クラスタ環境のみ) 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。

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 monCIPCLinks のコラムは次のとおりです。

名前	データ型	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である。
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である。
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 コ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 コラムが更新されてから経過した時間 (ミリ秒単位)。

monCIPCMesh

説明

(クラスタ環境のみ) 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。

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

monCIPCMesh のコラムは次のとおりです。

名前	データ型	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	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

(クラスタ環境のみ) 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。

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

monCLMObjectActivity のカラムは次のとおりです。

カラム名	データ型	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。

カラム名	データ型	説明
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。
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 (クラスタ環境のみ) 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。

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	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 (クラスタ環境のみ) 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。

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	データ型	説明
InstanceID	tinyint1	フェールオーバー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。
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		(クラスタ環境のみ) 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。
Status	varchar(30)	null	キャッシュのステータス。次のいずれか。 <ul style="list-style-type: none"> • Active • Pending/Active • Pending/Delete • Update Cache • Cache Create • Cache Delete • (Cluster Edition のみ) Cache Skip

名前	データ型	属性	説明
Type	varchar(30)	null	キャッシュのタイプ。次のいずれか。 <ul style="list-style-type: none">• Default• Mixed• Mixed, HK Ignore• Log Only• In-Memory Storage
CacheSize	int		キャッシュの合計サイズ(キロバイト単位)。
ReplacementStrategy	varchar(30)	null	キャッシュ置換方式。
APFReads	int	カウンタ	このデータ・キャッシュの非同期プリフェッチ (APF) 読み込み回数。
Overhead	int		キャッシュ・オーバーヘッド。

monDBRecovery

説明 (クラスタ環境のみ) 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。

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	データ型	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。
AnlTotRec	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。
AnlPhase1Recs	int4	フェーズ 1 リカバリ・プロセス内のログ・レコード数。
AnlPhase1RedoRecs	int4	フェーズ 1 リカバリで再実行するログ・レコード数。
AnlPhase2Recs	int4	フェーズ 2 リカバリ・プロセス内のログ・レコード数。
AnlPhase2RedoRecs	int4	フェーズ 2 リカバリで再実行するログ・レコード数。
AnlTotPages	int4	分析プロセス中の log scan getpage 要求の数。
AnlTotAPFWaited	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。
AnlTotIO	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	データ型	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。
AniStartTime	datetime(8)	分析パスの開始時刻。
AniEndTime	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 (クラスタ環境のみ) 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。

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	データ型	説明
DBID	int4	データベースのユニークな識別子。
InstanceID	int1	(クラスタ環境のみ) リカバリを実行したインスタンス。
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。

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 max messages、および deadlock pipe active の各設定パラメータを有効にしてください。

コラム

monDeadLock のコ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。

名前	データ型	属性	説明
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 <code>spid</code> 。
WaitSPID	smallint		ロックを待機しているプロセスの <code>spid</code> 。
WaitKPID	int		ロックを待機しているプロセスの <code>kpid</code> 。
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。
HeldClientAppName	varchar(30)	Null	ロックを保持しているアプリケーションによって設定された <code>clientapplname</code> プロパティの値。
HeldClientName	varchar(30)	Null	ロックを保持しているアプリケーションによって設定された <code>clientname</code> プロパティの値。
HeldClientHostName	varchar(30)	Null	ロックを保持しているアプリケーションによって設定された <code>clienthostname</code> プロパティの値。
HeldHostName	varchar(30)	Null	ロックを保持しているクエリを実行したアプリケーションが実行されているホスト・マシンの名前。
HeldNumLocks	int		<code>spid</code> を保持することで現在保持されているロックの数。
HeldProcDBName	varchar(30)	Null	デッドロックの発生時にストアード・プロシージャがブロッキング・プロセスを実行していたデータベースの名前(該当する場合)。
HeldProcedureName	varchar(30)	Null	デッドロックの発生時にブロッキング・プロセスが実行していたストアード・プロシージャの名前(該当する場合)。
HeldStmtNumber	int		ロックを保持している SQL 文の SQL バッチ内の文の番号。
ObjectDBName	varchar(30)	Null	データベースの名前。
ObjectID	int	Null	オブジェクトのユニークな識別子。
WaitAppName	varchar(30)	Null	ロックを待機しているアプリケーションの名前。

名前	データ型	属性	説明
WaitBatchID	int		ロック・タイムアウトの発生時にロックを待機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。
WaitClientApplName	varchar(30)	Null	ロックを待機しているアプリケーションによって設定された <i>clientapplname</i> プロパティの値。
WaitClientHostName	varchar(30)	Null	ロックを待機しているアプリケーションによって設定された <i>clienthostname</i> プロパティの値。
WaitClientName	varchar(30)	Null	ロックを待機しているアプリケーションによって設定された <i>clientname</i> プロパティの値。
WaitCommand	varchar(30)	Null	プロセスのカテゴリ、またはプロセスがブロックされてタイムアウトになったときに実行していたコマンド。
WaitContextID	int		ロックを待機しているプロセスが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。
WaitHostName	varchar(30)	Null	ロックを待機しているプロセスを実行するホストの名前。
WaitLineNumber	int		ロックを待機している SQL バッチまたはストアード・プロシージャ内の SQL 文の行番号。
WaitProcDBID	int		ロックを待機しているストアード・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)。
WaitProcDBName	varchar(30)	Null	ロックを待機しているストアード・プロシージャが存在するデータベースの名前 (該当する場合)。
WaitProcDBName	varchar(30)	Null	ロックを待機しているストアード・プロシージャが存在するデータベースの名前 (該当する場合)。
WaitProcedureID	int		ロックを待機しているストアード・プロシージャの ID (該当する場合)。
WaitProcedureName	varchar(30)	Null	ロックを待機しているストアード・プロシージャの名前 (該当する場合)。
WaitStmntNumber	int		ロックを待機している SQL バッチの行番号。
WaitTranName	varchar(255)	Null	ロックが要求されたトランザクションの名前。

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。

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		(クラスタ環境のみ) 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。

monDeviceSpaceUsage

説明

データベース・デバイスが割り当てられたファイル・システムに関する情報を提供する。空き領域の情報は、ファイル・システム・デバイスについてのみ表示される。ファイル・システム・サイズおよび空き領域の値は、ロー・デバイスに割り当てられたデータベース・デバイスの場合 `null`。

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

`monDeviceSpaceUsage` のカラムは次のとおりです。

名前	データ型	説明
InstanceID	tinyint	(クラスタ環境のみ) インスタンスの ID。
VDevNo	int	デバイスの仮想番号。
LogicalName	varchar(30)	デバイスの論理名。
PhysicalName	varchar(128)	デバイスの物理名。
DeviceSizeMB	int	デバイスのサイズ (メガバイト単位)。
FileSystemName	varchar(128)	ファイル・システムの名前。
FileSystemSizeMB	int	ファイル・システムのサイズ (メガバイト単位)。
FileSystemFreeMB	int	ファイル・システムで利用できる空き容量の総量 (メガバイト単位)。

monEngine

説明

Adaptive Server エンジンについての統計情報を提供します。

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。
ThreadID	int		エンジンに関連付けられているスレッドの ID。
InstanceID	int		(クラスタ環境のみ) 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	このエンジンがオペレーティング・システムに対して CPU を解放した回数。Adaptive Server をプロセス・モードで実行している場合は、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。Adaptive Server をスレッド・モードで実行している場合は、アイドル期間中の CPU 解放率は、 <code>alter thread pool .. idle timeout</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	(プロセス・モードのみ) 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。スレッドが I/O 完了のポーリングに使用されている場合は適用しない。
DiskIOPolled	int	カウンタ、リセット	(プロセス・モードのみ) 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。

名前	データ型	属性	説明
DiskIOCompleted	int	カウンタ、リセット	(プロセス・モードのみ) 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。
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。

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 max messages`、および `errorlog pipe active` 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	説明
SPID	smallint	セッション・プロセス識別子 (spid)。
InstanceID	int	(クラスタ環境のみ) 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。Adaptive Server バージョン 15.7 以降では値 99 を使用してスタック・トレースを示し、15.7 より前のバージョンでは値 0 を使用する。
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)	エラー・メッセージのテキスト属性は null。

monFailoverRecovery

説明

(クラスタ環境のみ) 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・メンバーシップ・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。

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	データ型	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。

monInmemoryStorage

説明 インメモリ・データベースのコンテンツを保存するように設定されているインメモリ・デバイスに関する情報を提供する。

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 monInmemoryStorage のカラムは次のとおり。

カラム名	データ型	説明
InstanceID	tinyint	(クラスタ環境のみ) 共有ディスク・クラスタ内のインスタンスの ID。
ID	int	このデバイスがバインドされているデータ・キャッシュの ID。
DeviceNum	int	デバイス番号。インメモリ・デバイスの場合は常に -1。
StartPage	int	このデバイスの最初のページのページ ID。
NumPage	int	このデバイスのページ数。
SizeKB	int	デバイスのサイズ (キロバイト単位)。
Name	varchar(30)	このデバイスのデータ・キャッシュの名前。
DeviceName	varchar(30)	インメモリ・ストレージ・デバイスの名前。
Type	varchar(30)	記憶領域のタイプ。常に “cache” に設定される。
Status	varchar(30)	デバイスのステータス。

monIOController

説明

I/O コントローラに関する情報を提供する。

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

monIOController のカラムは次のとおりです。

名前	データ型	属性	説明
InstanceID	int		(クラスタ環境のみ) 共有ディスク・クラスタ内のインスタンスの ID。
ControllerID	int		I/O コントローラの ID。
KTID	int		カーネル・タスクの ID。
EngineNumber	int		このコントローラを所有しているエンジン。
BlockingPolls	bigint	カウンタ	ブロックしているポーリングの数。
NonBlockingPolls	bigint	カウンタ	ブロックしていないポーリングの数。
EventPolls	bigint	カウンタ	イベントを返すポーリングの数。
NonBlockingEventPolls	bigint	カウンタ	イベントを返すブロックしていないポーリングの数。
FullPolls	bigint	カウンタ	最大数のイベントを返すポーリングの数。
Events	bigint	カウンタ	ポーリングされたイベントの数。
EventHWM	bigint	カウンタ	1 つのイベントで返される最大数のイベント。
Pending	int	カウンタ	保留中 I/O オペレーションの数。
Completed	bigint	カウンタ	完了した I/O オペレーションの数。
Reads	bigint	カウンタ	読み込みまたは受信オペレーションの数。
Writes	bigint	カウンタ	書き込みまたは送信オペレーションの数。
Deferred	bigint	カウンタ	延期または遅延 I/O オペレーションの数。
Type	varchar(30)		I/O コントローラ・タイプ。

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。

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		(クラスタ環境のみ) 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、System 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。

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		(クラスタ環境のみ) 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。
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		(クラスタ環境のみ) 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"> • 排他 • 共有 • 更新
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"> • ロー • ページ • 表 • アドレス
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。

名前	データ型	属性	説明
RowNumber	int	Null	LockLevel = 'ROW' のとき、ロックされるロー。
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。

monLockTimeout

説明

ロック・タイムアウトに関する情報を提供します。各ローはブロックされたロック要求が発生したオブジェクトと、ブロックされたプロセスとブロックしているプロセスの ID を示します。

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

カラム

monLockTimeout のカラムは次のとおりです。

名前	データ型	属性	説明
InstanceID	tinyint		(クラスタ環境のみ) クラスタ内のインスタンスの ID。
LockWaitPeriod	int		タイムアウトが発生する前にプロセスが待機する設定済み時間。
LockTimeoutLevel	varchar(20)	Null	タイムアウト・レベル。次のいずれか。 <ul style="list-style-type: none"> • DTM_SERVER • SERVER • SESSION • COMMAND • INVALID
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識別子。
ObjectDBName	varchar(30)	Null	オブジェクトが存在するデータベースの名前。
ObjectID	int		オブジェクトのユニークな識別子。
ObjectName	varchar(255)	Null	オブジェクトの名前。
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)。
ExpiredAtTime	datetime		ロックの期限切れ時刻。
HeldSPID	int		ロックを保持しているプロセスのサーバー・プロセス ID (spid)。
HeldKPID	int		ロックを保持しているプロセスのカーネル・プロセス ID (kpid)。
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。

名前	データ型	属性	説明
HeldHostName	varchar(30)	Null	ロックを保持しているクエリを実行したアプリケーションが実行されているホスト・マシンの名前。
HeldClientName	varchar(30)	Null	ロックを保持しているアプリケーションによって設定された clientname プロパティの値。
HeldClientApplName	varchar(30)	Null	ロックを保持しているアプリケーションによって設定された clientapplname プロパティの値。
HeldClientHostName	varchar(30)	Null	ロックを保持しているアプリケーションによって設定された clienthostname プロパティの値。
HeldTranName	varchar(255)	Null	ロックを取得したトランザクションの名前。
HeldCommand	varchar(30)	Null	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。
HeldFamilyID	int		ロックを保持している親プロセスの spid 。
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)。
HeldProcDBName	varchar(30)	Null	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名前 (該当する場合)。
HeldProcedureName	varchar(255)	Null	ロックの保持を引き起こしたストアド・プロシージャの名前 (該当する場合)。
HeldBatchID	int		ロック・タイムアウト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 SQL 文の SQL バッチ内の行番号。
HeldStmtNumber	int		ロックを保持している SQL 文の SQL バッチ内の文の番号。

名前	データ型	属性	説明
HeldLockType	varchar(20)	Null	ロックのタイプ。次のいずれか。 <ul style="list-style-type: none"> • 排他テーブル • 共有テーブル • 意図的排他 • 意図的共有 • 排他ページ • 共有ページ • 更新ページ • 排他ロー • 共有ロー • 更新ロー • ネクスト・キー • 排他アドレス • 共有アドレス • セマフォ
HeldNumLocks	int		spid を保持することで現在保持されているロックの数。
HeldNumTimeoutsCausedByTran	int		この保持トランザクションが原因となったタイムアウトの回数。
HeldNumTimeoutsCausedByLock	int		このロック・リソースが原因となったタイムアウトの回数。
HeldSourceCodeID	varchar(30)	Null	保持ロックが取得されたソース・コード内の位置 (内部使用のみ)。
WaitSPID	int		ロックを待機しているプロセスの spid。
WaitKPID	int		ロックを待機しているプロセスの kpid。
WaitUserName	varchar(30)	Null	ロックを要求したユーザの名前。
WaitAppName	varchar(30)	Null	ロックを待機しているアプリケーションの名前。
WaitHostName	varchar(30)	Null	ロックを待機しているプロセスを実行するホストの名前。
WaitClientName	varchar(30)	Null	ロックを待機しているアプリケーションによって設定された <i>clientname</i> プロパティの値。
WaitClientAppName	varchar(30)	Null	ロックを待機しているアプリケーションによって設定された <i>clientaplname</i> プロパティの値。

名前	データ型	属性	説明
WaitClientHostName	varchar(30)	Null	ロックを待機しているアプリケーションによって設定された <i>clienthostname</i> プロパティの値。
WaitTranName	varchar(255)	Null	ロックが要求されたトランザクションの名前。
WaitCommand	varchar(30)	Null	プロセスのカテゴリ、またはプロセスがブロックされてタイムアウトになったときに実行していたコマンド。
WaitFamilyID	int		ロックを待機している親プロセスの <i>spid</i> 。
WaitProcDBID	int		ロックを待機しているストアド・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)。
WaitProcDBName	varchar(255)	Null	ロックを待機しているストアド・プロシージャが存在するデータベースの名前 (該当する場合)。
WaitProcedureName	varchar(255)	Null	ロックを待機しているストアド・プロシージャの名前 (該当する場合)。
WaitBatchID	int		ロック・タイムアウトの発生時にロックを待機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。
WaitContextID	int		ロックを待機しているプロセスが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。
WaitLineNumber	int		ロックを待機している SQL バッチ内の SQL 文の行番号。
WaitStmtNumber	int		ロックを待機している SQL バッチの行番号。

名前	データ型	属性	説明
WaitLockType	varchar(30)	Null	<p>ロックのタイプ。次のいずれか。</p> <ul style="list-style-type: none"> • 排他テーブル • 共有テーブル • 意図的排他 • 意図的共有 • 排他ページ • 共有ページ • 更新ページ • 排他ロー • 共有ロー • 更新ロー • ネクスト・キー • 排他アドレス • 共有アドレス • セマフォ
WaitNumTimeoutsCausedByTran	int		トランザクションの待機が原因となったタイムアウトの回数。
WaitSourceCodeID	int		タイムアウトが発生し、待機ロック要求が行われたときのソース・コード内の位置 (内部使用のみ)。
HeldProcedureID	int		タイムアウトの発生時にブロッキング・プロセスが実行していたストアド・プロシージャのユニークなオブジェクト識別子。
WaitProcedureID	int		ロックを待機しているストアド・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)。

monLogicalCluster

説明 (クラスタ環境のみ) 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。

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	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	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 (クラスタ環境のみ) 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論理クラスタに対するすべての管理アクションを示します。

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

名前	データ型	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、releasing、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。次のいずれか。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時 (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時 (InstancesWaiting がゼロで、アクションのステータスが active から complete に変わった日時)。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・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、フェールオーバできない。

monLogicalClusterInstance

説明 (クラスタ環境のみ) 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。

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	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 (クラスタ環境のみ) 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。

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	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。次のいずれか。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。

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		(クラスタ環境のみ) 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。
PacketsSentMln	int		サーバが送信したパケットの数 (百万単位)。
PacketsReceivedMln	int		サーバが受信したパケットの数 (百万単位)。
BytesSentMB	int		サーバが送信したパケットの数 (メガバイト単位)。
BytesReceivedMB	int		サーバが受信したパケットの数 (メガバイト単位)。

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。

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

monOpenDatabases 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。
InstanceID	int		(クラスタ環境のみ) 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。 truncate_only または no_log を使用してトランザクションがダンプされる場合は、時刻は更新されない。

名前	データ型	属性	説明
PRUpdateCount	int	カウンタ	ベース・テーブルに対する insert、update、または delete によって発生した、事前計算済み結果セットの更新回数。
PRSelectCount	int	カウンタ	クエリ・プランの生成時に、オプティマイザがこのデータベースの事前計算済み結果セットを選択した回数。
PRRewriteCount	int	カウンタ	クエリ・プランの生成時に、オプティマイザがこのデータベースの事前計算済み結果セットが有効であると判断した回数。

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。

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**、および **object lockwait timing** の各設定パラメータを有効にしてください。

コラム monOpenObjectActivity 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。
InstanceID	int		(クラスタ環境のみ) 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。

名前	データ型	属性	説明
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	オプ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	int		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。
HkgcPending	int		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	int		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		(クラスタ環境のみ) オブジェクトごとに要求される物理ロックの数。
PhsycialLocksRetained	int		(クラスタ環境のみ) 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		(クラスタ環境のみ) 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		(クラスタ環境のみ) 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		(クラスタ環境のみ) インスタンスが物理ロック要求を待機した回数。

名前	データ型	属性	説明
PhysicalLocksPageTransfer	int		(クラスタ環境のみ) 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		(クラスタ環境のみ) 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		(クラスタ環境のみ) 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。
MaxPhysicalLockWaitTime	real		(クラスタ環境のみ) 物理ロックが付与されるまでこのオブジェクトが待機した最大回数。
AvgTransferReqWaitTime	int4		(クラスタ環境のみ) 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。
MaxTransferReqWaitTime	real		(クラスタ環境のみ) 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最大時間。
TotalServiceRequests	int4		(クラスタ環境のみ) 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		(クラスタ環境のみ) 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。
PagesTransferred	int4		(クラスタ環境のみ) 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		(クラスタ環境のみ) 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		(クラスタ環境のみ) 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。
MaxServiceTime	real		(クラスタ環境のみ) 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最大サービス時間。
AvgQueueWaitTime	real		(クラスタ環境のみ) このオブジェクトのバッファ転送を完了するために、待機に費やされた平均時間(ミリ秒単位)。
MaxQueueWaitTime	real		(クラスタ環境のみ) このオブジェクトのバッファ転送を完了するために、待機に費やされた最大時間(ミリ秒単位)。

名前	データ型	属性	説明
AvgTimeWaitedOnLocalUsers	int4		(クラスタ環境のみ) このインスタンスの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・マネージャが待機に費やした平均時間(ミリ秒)。
MaxTimeWaitedOnLocalUsers	real		(クラスタ環境のみ) このインスタンスの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・マネージャが待機に費やした最大時間(ミリ秒)。
AvgTransferSendWaitTime	int4		(クラスタ環境のみ) 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時間。
MaxTransferSendWaitTime	real		(クラスタ環境のみ) 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を完了するために費やした最大時間。
AvgIOServiceTime	int4		(クラスタ環境のみ) 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のために使用した平均時間。
MaxIOServiceTime	real		(クラスタ環境のみ) クラスタ・キャッシュ・マネージャがディスクへのページの書き込みに費やした最大時間。
AvgDowngradeServiceTime	int4		(クラスタ環境のみ) クラスタ・キャッシュ・マネージャが物理ロックをダウングレードするために使用した平均時間。
MaxDowngradeServiceTime	real		(クラスタ環境のみ) ページ上で物理ロックをダウングレードするためにタスクが待機に費やした最大時間。
SharedLockWaitTime	int	カウンタ、リセット、null	すべてのタスクが共有ロックの待機に費やした合計時間(ミリ秒単位)。
ExclusiveLockWaitTime	int	カウンタ、リセット、null	すべてのタスクが排他ロックの待機に費やした合計時間(ミリ秒単位)。
UpdateLockWaitTime	int	カウンタ、リセット、null	すべてのタスクが更新ロックの待機に費やした合計時間(ミリ秒単位)。
ObjectCacheDate	datetime	カウンタ、リセット、null	オブジェクトをキャッシュに追加した日付と時刻を示す。
PRSSelectCount	int	カウンタ、null	事前計算済み結果セットがクエリで使用された回数。
LastPRSSelectDate	datetime	null	事前計算済み結果セットがクエリで最後に使用された日付。

名前	データ型	属性	説明
PRSRewriteCount	int	カウンタ、 null	オプティマイザが、事前計算済み結果セットをクエリで使用することが有効であると判断した回数。オプティマイザは、より適した選択肢を発見したために事前計算済み結果セットを使用しない場合があります。
LastPRSRewriteDate	datetime	null	オプティマイザが、事前計算済み結果セットをクエリで使用することが有効であると最後に判断した日付。

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** 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。

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 `monOpenPartitionActivity` 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。
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。

名前	データ型	属性	説明
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	int		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。
HkgcPending	int		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 の再起動時にそれらのページのガーベジは収集されない。
HkgcOverflows	int		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		(クラスタ環境のみ) オブジェクトごとに要求される物理ロックの数。
PhysicalLocksRetained	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		(クラスタ環境のみ) 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		(クラスタ環境のみ) 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		(クラスタ環境のみ) インスタンスが物理ロック要求を待機した回数。
PhysicalLocksPageTransfer	int		(クラスタ環境のみ) 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。

名前	データ型	属性	説明
TransferReqWaited	int4		(クラスタ環境のみ) 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。
MaxPhysicalLockWaitTime	real		(クラスタ環境のみ) 物理ロックが付与されるまでこのオブジェクトが待機した最大回数。
AvgPhysicalLockWaitTime	int4		(クラスタ環境のみ) 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。
MaxTransferReqWaitTime	real		(クラスタ環境のみ) 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最大時間。
AvgTransferReqWaitTime	int4		(クラスタ環境のみ) 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		(クラスタ環境のみ) 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		(クラスタ環境のみ) 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。
PagesTransferred	int4		(クラスタ環境のみ) 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		(クラスタ環境のみ) 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		(クラスタ環境のみ) 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時間。
MaxServiceTime	real		(クラスタ環境のみ) 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最大時間。
AvgQueueWaitTime	int		(クラスタ環境のみ) このオブジェクトのバッファ転送を完了するために、待機に費やされた平均時間 (ミリ秒単位)。
MaxQueueWaitTime	int		(クラスタ環境のみ) このオブジェクトのバッファ転送を完了するために、待機に費やされた最大時間 (ミリ秒単位)。
AvgTimeWaitedOnLocalUsers	int4		(クラスタ環境のみ) 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・マネージャが待機に費やした平均サービス時間。

名前	データ型	属性	説明
MaxTimeWaitedOnLocalUsers	real		(クラスタ環境のみ) 目的のオブジェクトが別のプロセスで使用されているために、インスタンスのクラスタ・キャッシュ・マネージャが物理ロックの待機に費やした最大時間(ミリ秒単位)。
AvgTransferSendWaitTime	int4		(クラスタ環境のみ) 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。
MaxTransferSendWaitTime	real		(クラスタ環境のみ) 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を完了するために待機した最大時間。
AvgIOServiceTime	int4		(クラスタ環境のみ) 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。
MaxIOServiceTime	real		(クラスタ環境のみ) クラスタ・キャッシュ・マネージャがディスクへのページの書き込みに費やした最大時間。
AvgDowngradeServiceTime	int4		(クラスタ環境のみ) クラスタ・キャッシュ・マネージャが物理ロックをダウングレードするために使用した平均時間。
MaxDowngradeServiceTime	real		(クラスタ環境のみ) ページ上で物理ロックをダウングレードするためにタスクが待機に費やした最大時間。
ObjectCacheDate	datetime	カウンタ、リセット、null	オブジェクトをキャッシュに追加した日付と時刻を示す。
HkgcRequestsDcomp	int		ページ圧縮のためにキューに置かれたパーティションの合計データ・ページ数。
HkgcPendingDcomp	int		ページ圧縮を保留しているパーティションのデータ・ページ数。
HkgcOverflowsDcomp	int		ハウス・キーピング・キューが満杯なので、圧縮できなかったページの合計数。
IOSize1Page	int		各 IO (1 ページ) の IO 操作の数。
IOSize2Pages	int		各 IO (2 ページ) の IO 操作の数。
IOSize4Pages	int		各 IO (4 ページ) の IO 操作の数。
IOSize8Pages	int		各 IO (8 ページ) の IO 操作の数。

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。

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	(クラスタ環境のみ) 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 size 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。

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	(クラスタ環境のみ) 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。
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。

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	(クラスタ環境のみ) 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

(クラスタ環境のみ) 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。

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	データ型	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	データ型	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。
MinTimeCIPCSendCB	flt4	モジュールごとに cipc_sendcb に費やした最小時間。
AvgTimeCIPCSendCB	flt4	モジュールごとに cipc_sendcb に費やした平均時間。
MaxTimeCIPCSendCB	flt4	モジュールごとに cipc_sendcb 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。

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 <code>sysprocedures</code> 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		(クラスタ環境のみ) 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つのローが含まれています。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。

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		(クラスタ環境のみ) 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つのローが含まれています。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。

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		(クラスタ環境のみ) 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。

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。

コラム monProcess 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		(クラスタ環境のみ) 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)。

名前	データ型	属性	説明
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。
HostName	varchar(30)	Null	プロセスを開始したアプリケーションが実行されているホスト・マシンの名前。
ClientName	varchar(30)	Null	アプリケーションによって設定された <i>clientname</i> プロパティの値。
ClientHostName	varchar(30)	Null	アプリケーションによって設定された <i>clienthostname</i> プロパティの値。
ClientAppName	varchar(30)	Null	アプリケーションによって設定された <i>clientappliance</i> プロパティの値。

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。

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

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		(クラスタ環境のみ) 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)。ServerUserID の値は、syslogins.suid カラムに一致する。対応する名前を取得するには suser_name 関数を使用する。
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。この値は通常のキャッシュと tempdb ユーザ・ログ・キャッシュの合計です。
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。この値は通常のキャッシュと tempdb ユーザ・ログ・キャッシュの合計です。

名前	データ型	属性	説明
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)。この値は通常のキャッシュと tempdb ユーザ・ログ・キャッシュの合計です。
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)。この値は通常のキャッシュと tempdb ユーザ・ログ・キャッシュの合計です。
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。
HostName	varchar(30)	Null	クエリを実行したアプリケーションが実行されているホスト・マシンの名前。
Application	varchar(30)	Null	アプリケーションの名前。
ClientName	varchar(30)	Null	アプリケーションによって設定された <i>clientname</i> プロパティの値。
ClientHostName	varchar(30)	Null	アプリケーションによって設定された <i>clienthostname</i> プロパティの値。
ClientApplName	varchar(30)	Null	アプリケーションによって設定された <i>clientapplname</i> プロパティの値。

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 \(200 ページ\)](#)」を参照してください。

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 monProcessLookup 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		(クラスタ環境のみ) 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。
ClientName	varchar(30)	Null	アプリケーションによって設定された <i>clientname</i> プロパティの値。
ClientHostName	varchar(30)	Null	アプリケーションによって設定された <i>clienthostname</i> プロパティの値。
ClientAppName	varchar(30)	Null	アプリケーションによって設定された <i>clientappliance</i> プロパティの値。

clientname、*clienthostname*、*clientappliance* の設定には `set` コマンドを使用します。『リファレンス・マニュアル：コマンド』を参照してください。

monProcessMigration

説明 (クラスタ環境のみ) 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。

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 monProcessMigration のカラムは次のとおりです。

名前	データ型	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。
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		(クラスタ環境のみ) 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。

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		(クラスタ環境のみ) 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。

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 **statement statistics active** の各設定パラメータを有効にしてください。

コラム monProcessProcedures 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		(クラスタ環境のみ) 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。
StmtNumber	int		現在実行中の文。
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)。
ExecutionCount	int	カウンタ	プロシージャ・キャッシュに保持されたストアド・プロシージャのこのインスタンスが Adaptive Server により実行された回数。
CPUTime	int	カウンタ	Adaptive Server でプロシージャ・キャッシュに保持されたこのストアド・プロシージャのインスタンスの実行に費やした CPU 時間 (ミリ秒単位)。
ExecutionTime	int	カウンタ	Adaptive Server でプロシージャ・キャッシュに保持されたこのストアド・プロシージャのインスタンスの実行に費やした時間 (ミリ秒単位)。

名前	データ型	属性	説明
PhysicalReads	int	カウンタ	プロシージャ・キャッシュに保持されたこのストアド・プロシージャのインスタンスにより実行された物理読み込みの回数。
LogicalReads	int	カウンタ	プロシージャ・キャッシュに保持されたこのストアド・プロシージャのインスタンスにより実行された論理読み込みの回数。
PhysicalWrites	int	カウンタ	プロシージャ・キャッシュに保持されたこのストアド・プロシージャのインスタンスにより実行された物理書き込みの回数。
PagesWritten	int	カウンタ	プロシージャ・キャッシュに保持されたこのストアド・プロシージャのインスタンスにより読み込まれたページ数。

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。

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、max SQL text monitored、および SQL batch capture 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		(クラスタ環境のみ) 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。

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`、`per object 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		(クラスタ環境のみ) 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。

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、wait event timing、および process wait events 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		(クラスタ環境のみ) 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。

『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。

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		(クラスタ環境のみ) 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 <code>spid</code> (ワーカー・プロセスの場合)。

monRepLogActivity

説明 Replication Agent で更新されたモニタ・カウンタから情報を収集します。
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 monRepLogActivity のコラムは次のとおりです。

名前	データ型	説明
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。
SPID	int	セッション・プロセス識別子。
InstanceID	tinyint	(クラスタ環境のみ) 共有ディスク・クラスタ内のインスタンスの ID。
LogRecordsScanned	int	スキャンされたログ・レコードの合計数。
LogRecordsProcessed	int	処理されたログ・レコードの合計数。
NumberOfScans	int	実行されたスキャンの総数。
TotalTimeForLogScans	bigint	スキャナ・スレッドがログのスキャンに費やした合計時間。
LongestTimeForLogScans	bigint	1 回のスキャンに費やした最長時間。
AvgTimeForLogScans	bigint	ログのスキャンに費やした平均時間。
Updates	int	処理された update の総数。
Inserts	int	処理された insert の総数。
Deletes	int	処理された delete の総数。
StoredProcedures	int	処理されたストアド・プロシージャの総数。
SQLStatements	int	処理された SQL 文の総数。
DDL	int	処理された DDL ログ・レコードの合計数。
Writetext	int	writetext コマンドで処理されたログ・レコードの総数。
LobColumns	int	off-ow で、大規模なオブジェクト・コラムを持つテーブルに対して処理された、DML ログ・レコードの総数。
CLRs	int	処理された CLR の総数。
Checkpoints	int	処理された checkpoint の総数。
BeginTransaction	int	処理された begin transaction の総数。
CommitTransaction	int	処理された commit transaction の総数。
AbortedTransaction	int	処理された aborted transaction の総数。
PreparedTransaction	int	準備状態にあるトランザクションの総数。
DelayedCommit	int	処理された delayed commit の総数。
MaintenanceUserTransaction	int	メンテナンス・ユーザがオープンしたトランザクションの総数。

名前	データ型	説明
NumberOfLogExtentions	int	RepAgent がトランザクションの拡張を待っていた合計回数。
TotalTimeOfLogExtentions	bigint	RepAgent がログの拡張を待っていたミリ秒 (ms) 単位の合計時間。
LongestTimeOfLogExtentions	bigint	RepAgent がログの拡張を待っていたミリ秒 (ms) 単位の最長時間。
AvgTimeOfLogExtentions	bigint	RepAgent がログの拡張を待っていたミリ秒 (ms) 単位の平均時間。
MaxHashSchemaSize	int	ハッシュ・スキーマ・キャッシュの最大サイズ。
NumberOfSchemasReused	int	再利用されたスキーマの総数。
NumberOfSchemaFwdLookup	int	前方ルックアップ・スキーマの総数。
TotalTimeOfSchemaFwdLookup	bigint	前方スキャンの実行に費やした合計時間 (ms)。
LongestTimeOfSchemaFwdLookup	bigint	前方スキャンの実行に費やした最長時間 (ms)。
AvgTimeOfSchemaFwdLookup	bigint	前方スキャンの実行に費やした平均時間 (ms)。
NumberOfSchemaBckwLookup	int	後方ルックアップ・スキーマの総数。
TotalTimeOfSchemaBckwLookup	bigint	後方ルックアップ・スキーマの実行に費やした合計時間。
LongestTimeOfSchemaBckwLookup	bigint	後方スキャンの実行に費やした最長時間 (ms)。
AvgTimeOfSchemaBckwLookup	bigint	後方スキャンの実行に費やした平均時間 (ms)。
NumberOfMempoolAllocates	int	プールが割り当てる RepAgent の総数。
NumberOfMempoolFrees	int	メモリ・プールが解放する RepAgent の総数。
MempoolCurrentSize	int	RepAgent メモリ・プールの現在のサイズ。
MempoolHighUsage	int	RepAgent メモリ・プールの高い使用率。
DBName	varchar(30)	タスクがスキャンするデータベースの名前。

monRepScanners

説明 Rep Agent スキャナ・タスクが費やす時間に関する情報を提供します。
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 monRepScanners のカラムは次のとおりです。

説明	データ型	説明
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。
SPID	int	セッション・プロセス識別子。
InstanceID	tinyint	(クラスタ環境のみ) 共有ディスク・クラスタ内のインスタンスの ID。
EngineBinding	int	このタスクをバウンドするエンジンの数 (スレッド・モードには適用できません)。
LogRecordsScanned	int	スキャンされたログ・レコードの合計数。
LogrecordsProcessed	int	処理されたログ・レコードの合計数。
NumberOfTruncPointRequested	int	RepAgent が Replication Server に新しいトランケーション・ポイントを要求した合計回数。
NumberOfTruncPointMoved	int	RepAgent がセカンダリ・トランケーション・ポイントを移動させた合計回数。
DBName	varchar(30)	このタスクがスキャンするデータベースの名前。
Status	varchar(30)	タスクの現在のステータス。
SleepStatus	varchar(30)	スリープしているものがある場合、スリープの現在のステータス。
StartMarker	varchar(30)	このスキャナのログ内の起動マーカ。
EndMarker	varchar(30)	このスキャナのログ内の終了マーカ。
CurrentMarker	varchar(30)	このスキャナのログ内の現在のマーカ。
OldestTransaction	varchar(30)	最も古いオープン・トランザクション。

monRepScannersTotalTime

説明 Rep Agent スキャナ・タスクが費やす時間に関する情報を提供します。
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 monRepScannersTotalTime のテーブルは次のとおりです。

名前	データ型	説明
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。
SPID	int	セッション・プロセス識別子。
InstanceID	tinyint	(クラスタ環境のみ) 共有ディスク・クラスタ内のインスタンスの ID。
LogRecProcessed	bigint	スキャナ・スレッドによって処理されるログ・レコードの総数。
BytesPacked	bigint	スキャナ・スレッドによってパックされるバイト数。
TotalTime	bigint	スキャナ・スレッドが使用した合計時間。
MRPBootstrapTime	bigint	マルチパス複写ブートストラップ・サイクルを完了するために必要な合計時間 (ミリ秒単位)。
ScanTime	bigint	スキャンの実行に費やした合計時間。
ProcessTime	bigint	ログ・レコードの処理に費やした合計時間。
SchemaLookupsTime	bigint	RepAgent キャッシュ内のオブジェクトのスキーマを探すために費やした合計時間。
PackTime	bigint	LTL のパッキングの実行に費やした合計時間。
QueueingTime	bigint	LTL バケットのキューイングに費やした合計時間。
HashBindingSize	bigint	オブジェクトのバインド情報を保持しているハッシュ・バインド・テーブル内のバケットの総数。
HashBindingEntries	bigint	RepAgent がブート・ストラップされたときのパスにバインドされるオブジェクトの総数。
HashBindingCollisions	bigint	ハッシュ・バインド・テーブル内で使用された衝突チェーンの最大長。
YieldsOnFullQueue	bigint	フル・キューでスキャナが開放する総数。
WaitsOnSenderThread	bigint	送信者スレッド上の待機の総数。
WaitTimeOnSenderThread	bigint	送信者スレッドでの待機の合計時間 (ミリ秒単位)。
LongestWaitOnSenderThread	bigint	送信者スレッドでの待機の最長時間 (ミリ秒単位)。

monRepSenders

説明

Rep Agent の送信者タスクに関する処理情報を提供します。

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

monRepSenders のカラムは次のとおりです。

名前	データ型	説明
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。
SPID	int	セッション・プロセス識別子。
InstanceID	tinyint	(クラスタ環境のみ) 共有ディスク・クラスタ内のインスタンスの ID。
EngineBinding	int	このタスクをバウンドするエンジンの数 (スレッド・モードには適用できません)。
MessageQueueSize	int	メッセージ・キューの最大サイズ。
MessagesInQueue	int	メッセージ・キュー内のメッセージの総数。
NumberOfScannerYields	int	フル・キューでスキャナが開放した合計時間。
NumberOfScannerSleeps	int	フル・キューでスキャナがスリープした合計時間。
NumberOfBytesSent	int	送信されたバイトの合計数。
LastRepServerError	int	Replication Server の最新のエラー。
NumberOfRetries	int	コネクションに対するリトライの合計回数。
SleepsOnEmptyQueue	int	空のメッセージ・キューに費やしたスリープの総数。
NumberOfQueueFlushes	int	送信者がキューをフラッシュした合計回数。
SleepTimeOnEmptyQueue	int	空のキュー内のスリープで費やした合計時間 (ms)。
LongestSleepTimeOnEmptyQueue	int	空のキュー内のスリープで費やした最長時間 (ms)。
MaxQueueSize	int	到達していないキューの最大サイズ。
DBName	varchar(30)	タスクがスキャンするデータベースの名前。
Dataserver	varchar(30)	Replication Server に接続するときに使用するデータ・サーバ名。
ReplicationServer	varchar(30)	Replication Server に接続するときに使用する Replication Server 名。
Username	varchar(30)	Replication Server に接続するときに使用するユーザ名。
Status	varchar(30)	このタスクの現在のステータス。
SleepStatus	varchar(30)	スリープしているものがある場合、スリープの現在のステータス。

monSpinlockActivity

説明 スピンロック・アクティビティの統計情報を提供する。
このモニタリング・テーブルでデータを収集するには、`enable spinlock monitoring` 設定パラメータを有効にしてください。

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

名前	データ型	説明
SpinlockName	varchar(255)	スピンロックの名前。
Grabs	bigint	このスピンロックのグラブの数。
Spins	bigint	このスピンロックのスピン数。
Waits	bigint	このスピンロックの待機の数。
OwnerPID	int	現在の所有者のプロセス識別子。
LastOwnerPID	int	前の所有者のプロセス識別子。
Contention	real	スピンロック競合 (パーセンテージ)。
InstanceID	tinyint	(クラスタ環境のみ) 共有ディスク・クラスタ内のインスタンスの ID。

monSQLRepActivity

説明

すべてのオープン・オブジェクトに対して正常に複製された SQL 文の統計情報を提供します。

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

monSQLRepActivity のカラムは次のとおりです。

名前	データ型	説明
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。
ObjectID	int	モニタしているオブジェクトの ID。
InstanceID	tinyint	(クラスタ環境のみ) 共有ディスク・クラスタ内のインスタンスの ID。
DBName	varchar(30)	アクティビティをモニタしているオブジェクトを含むデータベースの名前。
ObjectName	varchar(30)	アクティビティをモニタしているオブジェクトの名前。
UpdateStmts	int	SQL として複製された update 文の数。
InsertSelectStmts	int	SQL として複製された insert 文と select 文の数。
DeleteStmts	int	SQL として複製された delete 文の数。
SelectIntoStmts	int	SQL として複製された select into 文の数。
RowsThreshold	int	文の影響を受けるロー番号範囲の下限。

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。

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		(クラスタ環境のみ) 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指定する。
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。

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**、**enable stmt cache monitoring**、および **statement cache size** の各設定パラメータを有効にしてください。

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

名前	データ型	属性	説明
InstanceID	tinyint		(クラスタ環境のみ) 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

(クラスタ環境のみ) 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。

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

`monSysLoad` のカラムは次のとおりです。

名前	データ型	説明
<code>InstanceID</code>	<code>tinyint</code>	クラスタ内でのインスタンスの ID。
<code>EngineNumber</code>	<code>smallint</code>	このローが属するエンジン。
<code>SteadyState</code>	<code>real</code>	<code>Adaptive Server</code> が起動してからのこの統計の平均値。
<code>Avg_1min</code>	<code>real</code>	この統計の 1 分間の移動平均。
<code>Avg_5min</code>	<code>real</code>	この統計の 5 分間の移動平均。
<code>Avg_15min</code>	<code>real</code>	この統計の 15 分間の移動平均。
<code>Max_1min</code>	<code>real</code>	起動時からの 1 分間最大平均。
<code>Max_5min</code>	<code>real</code>	起動時からの 5 分間最大平均。
<code>Max_15min</code>	<code>real</code>	起動時からの 15 分間最大平均。
<code>Max_1min_Time</code>	<code>datetime</code>	<code>Max_1min</code> が発生した <code>datetime</code> 。
<code>Max_5min_Time</code>	<code>datetime</code>	<code>Max_5min</code> が発生した <code>datetime</code> 。
<code>Max_15min_Time</code>	<code>datetime</code>	<code>Max_15min</code> が発生した <code>datetime</code> 。
<code>Statistic</code>		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
<code>Sample</code>	<code>float</code>	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)。
<code>Peak</code>	<code>float</code>	インスタンスが起動してからの <code>Sample</code> の最大値 (<code>Sample</code> ピーク値)。
<code>Peak_time</code>	<code>datetime</code>	<code>Peak</code> 値に達した日付と時刻。
<code>StatisticID</code>	<code>int</code>	この統計の固定識別子。ローカライズされた <code>Statistic</code> 名ではなく、固定の <code>StatisticID</code> 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・プランの履歴を提供します。
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。

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 max messages`、および `plan text pipe active` 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		(クラスタ環境のみ) 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 <code>PlanText</code> 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。

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列してください。

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。

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`、`sql text pipe max messages`、および `sql text pipe active` 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		(クラスタ環境のみ) 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列してください。

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。

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、per object statistics active、statement pipe max messages、および statement pipe 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。

カラム

monSysStatements 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		(クラスタ環境のみ) 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。

名前	データ型	属性	説明
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、このカラムはゼロ (0)。
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。

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		(クラスタ環境のみ) 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。

『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。

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 `monSysWorkerThread` のコラムは次のとおりです。

名前	データ型	属性	説明
InstanceID	int		(クラスタ環境のみ) 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 (<code>number of worker processes</code> 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。 <code>max parallel degree</code> 設定オプションの現在の <code>Run Value</code> 。
MaxScanParallelDegree	int		スキャンに使用可能な最大並列度。 <code>max scan parallel degree</code> 設定オプションの現在の <code>Run Value</code> 。

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。

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 (カラムの測定単位を含む)。
Language	varchar(30)		このカラムにより、Adaptive Server で Description カラムと Label カラムの値を返す言語を指定できる。 デフォルトでは英語を使用する。クエリでは ISO-639 および ISO-3166 命名規則を使用する。
Label	varchar(50)		カラム内のデータの説明。アプリケーションのユーザ・インタフェースでは、実際のカラム名の代わりにこれらの値を使用できる。

Indicators 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
```

monTableCompression

説明 テーブルの圧縮履歴を含みます。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 `monTableCompression` のカラムは次のとおりです。

名前	データ型	属性	説明
InstanceID	tinyint		(Cluster Edition のみ) サーバ・インスタンス ID。
DBID	int		このテーブルが転送されたデータベース ID。
ObjectID	int		圧縮されたオブジェクトの ID。
PartitionID	int		圧縮されたパーティションの ID。
CompRowInserted	bigint	カウンタ	挿入された圧縮ローの数。
CompRowUpdated	bigint	カウンタ	更新された圧縮ローの数。
CompRowForward	bigint	カウンタ	更新から転送された、圧縮されたローの数。
CompRowScan	bigint	カウンタ	アクセスされた圧縮ローの数。
RowDecompressed	bigint	カウンタ	圧縮解除されたローの数。
RowPageDecompressed	bigint	カウンタ	ロー圧縮をするために圧縮解除されたページ圧縮ローの数。
ColDecompressed	bigint	カウンタ	圧縮解除されたカラムの数。
RowCompNoneed	int	カウンタ	圧縮ローの長さが通常のロー長を超えているために圧縮されないローの数。
PageCompNoneed	bigint	カウンタ	Adaptive Server が辞書またはインデックスを生成できないためページレベルの圧縮には適さないページ数。
PagesCompressed	bigint	カウンタ	ページレベルで圧縮されるページ数。
AvgBytesSavedPageLevel	bigint	カウンタ	ページレベル圧縮によって保存されたバイト数。
TableName	varchar	NULL	圧縮されたテーブルの名前。

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。

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・テーブルについて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。

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。
Columns	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		<p>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)。</p> <p>Indicators 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る。値が 1 の場合、テーブルは履歴テーブルである。</p> <p>すべての履歴テーブルを表示するには、以下を実行する。</p> <pre>Select TableName from master..monTables where Indicators & 1 != 0</pre>
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。
Language	varchar(30)		<p>このカラムにより、Adaptive Server で Description カラムの値 を返す言語を指定できる。</p> <p>デフォルトでは英語を使用する。クエリでは ISO-639 および ISO-3166 命名規則を使用する。</p>

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。

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

monTableTransfer のコラムは次のとおりです。

名前	データ型	属性	説明
InstanceID	tinyint	NOT NULL	(クラスタ環境のみ) 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。

monTask

説明

Adaptive Server のスレッド・モード専用で、各タスクについて1つローを含みます。

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

monTask 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	クラスタ内でのインスタンスの ID。
KTID	int		カーネル・タスクの ID。
ThreadPoolID	int	NULL	このタスクが割り当てられているスレッド・プールの ID。
ThreadID	int		このタスクを実行するスレッドの ID。
KPID	int	NULL	Adaptive Server のカーネル・プロセス ID (KPID)。
SPID	int		セッション・プロセス識別子 (spid)。
Name	varchar(30)		タスクの名前。
ThreadPoolName	varchar(30)	NULL	このタスクが割り当てられているスレッド・プールの名前。

monTempdbActivity

説明

(クラスタ環境のみ) 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。

monThread

説明

Adaptive Server のスレッド・モード専用。各スレッドについて1つローを含みます。

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

monThread のカラムは次のとおりです。

名前	データ型	説明
InstanceID	tinyint	クラスタ内でのインスタンスの ID。
ThreadID	int	スレッド・プールの ID。
KTID	int	内部カーネル・スレッドの ID。
OSThreadID	int	オペレーティング・システム・スレッドの ID。
AltOSThreadID	int	代替オペレーティング・システム・スレッドの ID (プラットフォームにより、ライトウェイト・プロセス (LWP) ID になる場合もある)。
ThreadPoolID	int	スレッド・プールの ID。
State	varchar(30)	スレッドの現在のステータス。
ThreadAffinity	int	このスレッドが結び付いている CPU の数。
ThreadPoolName	varchar(30)	スレッド・プールの名前。
TaskRuns	bigint	このスレッドが実行したタスクの数。
TotalTicks	bigint	このスレッドの合計チック数。
IdleTicks	bigint	このスレッドがアイドルであった期間の合計チック数。
SleepTicks	bigint	このスレッドがスリープ状態にあった期間の合計チック数。
BusyTicks	bigint	このスレッドがビジーであった期間の合計チック数。
UserTime	bigint	スレッド・ユーザ CPU 時間の合計 (ミリ秒単位)。
SystemTime	bigint	スレッド・システム CPU 時間の合計 (ミリ秒単位)。
MinorFaults	bigint	マイナー・ページ・フォールトの総数。Windows では値は 0。
MajorFaults	bigint	メジャー・ページ・フォールトの総数。Windows では値は 0。
VoluntaryCtxtSwitches	bigint	自発的な OS コンテキストの切り替えの総回数。Windows では値は 0。
NonVoluntaryCtxtSwitches	bigint	自発的でない OS コンテキストの切り替えの総回数。Windows では値は 0。

monThreadPool

説明 Adaptive Server のスレッド・モード専用。各プールについて 1 つローを含みます。

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 monThreadPool のカラムは次のとおりです。

名前	データ型	説明
ThreadPoolID	int	スレッド・プールの ID。
Size	int	スレッド・プール内のスレッドの数。
TargetSize	int	必要なサイズ (プール・サイズを変更するときだけの Size とは異なる)。
Tasks	int	スレッド・プールに割り当てられたタスクの数。
ThreadPoolName	varchar(30)	スレッド・プールの名前。
ThreadPoolDescription	varchar(255)	(オプション) スレッド・プールの説明。
Type	varchar(30)	スレッド・プールのタイプ、エンジン (多重化) または実行完了 (RTC)。
IdleTimeout	int	アイドル状態に移行する前に、このプール内で実行可能なタスクを検索する時間 (マイクロ秒単位)。

monWaitClassInfo

説明

すべての待機クラスのテキストによる説明を提供します(たとえば、ディスクの読み込み完了を待機する場合)。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。

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。
Language	varchar(30)		このカラムにより、Adaptive Server で Description カラムの値を返す言語を指定できる。 デフォルトでは英語を使用する。クエリでは ISO-639 および ISO-3166 命名規則を使用する。

monWaitEventInfo

説明 `monSysWaits` テーブルおよび `monProcessWaits` テーブルでレポートされる待機状態のテキストによる説明を提供します。

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。
Language	varchar(30)		このカラムにより、Adaptive Server で Description カラムの値を返す言語を指定できる。 デフォルトでは英語を使用する。クエリでは ISO-639 および ISO-3166 命名規則を使用する。

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 (クラスタ環境のみ) 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。

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	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

(クラスタ環境のみ) 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。monWorkload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。

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

monWorkloadPreview のコラムは次のとおりです。

名前	データ型	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 (クラスタ環境のみ) 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。

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 `monWorkloadProfile` のカラムは次のとおりです。

名前	データ型	説明
ProfileID	smallint	負荷プロファイル ID。
ConnectionsWeight	tinyint	<code>active connections</code> 測定基準に関連付けられている重み付け。
CpuWeight	tinyint	<code>cpu utilization</code> 測定基準に関連付けられている重み付け。
RunQueueWeight	tinyint	<code>run queue</code> 測定基準に関連付けられている重み付け。
IoLoadWeight	tinyint	<code>io load</code> 測定基準に関連付けられている重み付け。
EngineWeight	tinyint	<code>engine deficit</code> 測定基準に関連付けられている重み付け。
UserWeight	tinyint	<code>user metric</code> 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 (クラスタ環境のみ) 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。

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	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。

monWorkQueue

説明

ワーク・キューに関する情報を提供します。

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

monWorkloadQueue のカラムは次のとおりです。

名前	データ型	説明
InstanceID	tinyint	(クラスタ環境のみ) 共有ディスク・クラスタ内のインスタンスの ID。
CurrentLength	int	キューに入っている項目の現在の数。
MaxLength	int	キューに入っている項目の最大数。
TotalRequests	int	要求の合計数。
QueuedRequests	int	別の要求の終了を待機した要求の合計数。
WaitTime	int	要求が待機した時間 (ミリ秒単位)。
Name	varchar(30)	ワーク・キューの名前。

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、プラットフォーム用の『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 PCI Bridge により使用される実行時ディスパッチ・モデルの実行時システム呼び出し設定情報を含みます。

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

数字

2 フェーズ・コミット
probe プロセス 60

A

allow updates to system tables パラメータ 9

D

dbid カラム、*sysusages* テーブル 108
DTX パティシパント 32

E

ESP。「拡張ストアド・プロシージャ」参照 47

G

grant コマンド
sysprotects テーブル 80

I

identity 値 5
ID、サーバ役割
sysroles テーブル 89

J

Java 項目
sysjars テーブル 51
sysxtypes テーブル 111

M

master データベース
システム・テーブル 1-2
monCachedObject テーブル 132
monCachedProcedures テーブル 134
monCachedStatement テーブル 135
monCachePool テーブル 133
monCIPC テーブル 138
monCIPCEndpoints テーブル 139
monCIPCLinks テーブル 140
monCIPCMesh テーブル 141
monCLMObjectActivity テーブル 143
monClusterCacheManager テーブル 145
monCMSFailover テーブル 146
monDataCache テーブル 147, 149
monDBRecoveryLRTypes テーブル 151
monDeadLock テーブル 152, 157
monDeviceIO テーブル 155
monErrorLog テーブル 159, 160
monIOController テーブル 162
monIOQueue テーブル 163
monLicense table 164
monLocks テーブル 165, 204
monLogicalCluster テーブル 172
monLogicalClusterAction テーブル 174
monLogicalClusterInstance テーブル 175
monLogicalClusterRoute テーブル 176
monNetworkIO テーブル 177
monOpenDatabases テーブル 178
monOpenObjectActivity テーブル 180
monOpenPartitionActivity テーブル 185
monPCIBridge テーブル 190, 191
monPCISlots テーブル 192
monPCM テーブル 193
monProcedureCache テーブル 195
monProcedureCacheMemoryUsage テーブル 196
monProcedureCacheModuleUsage テーブル 197
monProcess テーブル 198
monProcessActivity テーブル 200

monProcessLookup テーブル 202, 203
monProcessObject テーブル 205
monProcessProcedures テーブル 206
monProcessSQLText テーブル 208
monProcessStatement テーブル 209
monProcessWaits テーブル 210
monProcessWorkerThread テーブル 211
monState テーブル 220
monStatementCache テーブル 221
monSysLoad テーブル 222
monSysPlanText テーブル 223
monSysSQLText テーブル 224
monSysStatement テーブル 225
monSysWaits テーブル 227
monSysWorkerThread テーブル 228
monTableColumns テーブル 229
monTableCompression テーブル 231
monTableParameters テーブル 232
monTables テーブル 233
monTableTransfer テーブル 234
monTempdbActivity テーブル 236
monWaitClassInfo テーブル 239
monWaitEventInfo テーブル 240
monWorkload テーブル 241
monWorkloadPreview テーブル 242
monWorkloadProfile テーブル 243
monWorkloadRaw テーブル 244
monWorkQueue テーブル 245

O

OAM (オブジェクト・アロケーション・マップ) ページ数 98

P

probe プロセス、2 フェーズ・コミットの設定 60

R

revoke コマンド
sysprotects テーブル 80

S

segmap カラム、*sysusages* テーブル 108
segment カラム、*syssegments* テーブル 91
spid 番号 77
syslogshold 62
sysaudits テーブル 18
suid (サーバ・ユーザ ID)
sysalternates テーブル・リスト 11
syslogins テーブル・リスト 58
sybdiagdb データベース 4
syblicenseslog テーブル 4, 10
sybpcidb データベース
pca_jre_arguments 248
pca_jre_directives 249
pci_arguments 250
pci_directives 251
pci_slot_syscalls 253
pci_slotinfo 252
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 3
sysalternates テーブル 11
sysaltusages テーブル 12
sysattributes テーブル 14–16
sysauditoptions テーブル 17
sysaudits_01 ~ sysaudits_08 テーブル 18–19
syscharsets システム・テーブルの文字セット 23
syscharsets テーブル 23
syscolumns テーブル 24–26
syscomments テーブル 27–28
sysconfigures テーブル 29–30
sysconstraints テーブル 31
syscoordinations テーブル 32
syscurconfigs テーブル 33
sysdatabases テーブル 36–39
sysdepends テーブル 40
sysdevices テーブル 41–42
sysengines テーブル 45
sysgams テーブル 46
sysindexes テーブル 47–49
sysinstances システム・テーブル 50
sysjars テーブル 51
syskeys テーブル 52
syslanguages テーブル 53

syslisteners テーブル 54
syslocks テーブル 55–56
sysloginroles テーブル 57
syslogins テーブル 58–60
syslogs テーブル 61
 無限ループ (変更の場合) 61
syslogshold テーブル 62
sysmessages テーブル 63
sysmonitors テーブル 64
sysobjects テーブル 65–69
syspartitionkeys テーブル 72
sysprocedures テーブル 76
sysprocesses テーブル 77–79
sysprotects テーブル 80–82
sysquerymetrics テーブル 83
sysqueryplans テーブル 85
sysreferences テーブル 86
sysremotelogins テーブル 87
sysresourcelimits テーブル 88
sysroles テーブル 89
syssecmechs テーブル 90
syssegments テーブル 91
sysservers テーブル 92–93
sysessions テーブル 94
sysrvroles テーブル 96
sysstatistics テーブル 97
systabstats テーブル 98–99
systhresholds テーブル 100
systemranges テーブル 102
systransactions テーブル 103–104
systypes テーブル 105–107
sysusages テーブル 108
sysusermessages テーブル 109
sysusers テーブル 110
 sysalternates テーブル 11
sysxtypes テーブル 111

T

tempdb データベース
 システム・テーブル・エントリ 65–69
 timestamp カラム 5

U

us_english 言語 53

あ

アーカイブ・データベースへのアクセス
 sysaltusages テーブル 12
 スクラッチ・データベース 12
 アロケーション・ユニット
 sysusages テーブル 108
 暗号化
 役割パスワード 96
 ユーザ・パスワード 58

い

依存性、データベース・オブジェクト
 sysdepends テーブル 40
 インデックス
 システム・テーブル・エントリ 47
 インデックス・ページ
 数 98

え

英語、U.S。「us_english 言語」参照
 エイリアス、言語
 syslanguages テーブル 53
 エイリアス、ユーザ
 sysalternates テーブル 11
 エラー・メッセージ
 システム・テーブル・エントリ 63
 エンジン
 sysengines テーブル 45
 システム・テーブル・エントリ 45

か

階層

データ型 105

外部キー

syskeys テーブル 52

拡張ストアド・プロシージャ

システム・テーブル・エントリ 27, 65–69

数(量)

OAM ページ 98

インデックス・リーフ・ページ 98

インデックス・レベル 98

削除されたロー 98

転送されたロー 98

ページ 98

ロー 98

カラム

予約済み 8

監査

sysauditoptions テーブル 17

sysaudits_01 ~ *sysaudits_08* テーブル 18, 248

き

キー、テーブル

syskeys テーブル 52

共通キー

syskeys テーブル 52

く

クライアント

フェールバック中の削除 94

グループ

sysusers テーブル・エントリ 110

グローバル・アロケーション・マップのページ 46

グローバル監査オプション、*sysauditoptions* システム・テーブル 17

け

言語、代替

syslanguages テーブル 53

システム・テーブル・エントリ 53

検索

オブジェクトの依存性 40

オブジェクトの定義 27, 76

言語 53

制約 31

設定パラメータ 29, 33

データ型 105

データベース ID 36

データベース・オブジェクト 65

データベース設定 36

データベース内のユーザ 110

データベース名 36

デバイス名 41

パーミッション情報 80

文字セット 23

役割 89

ユーザ・エイリアス 11

リソース制限 88

検査制約

sysconstraints テーブル 31

システム・テーブル・エントリ 65–69, 76

こ

高可用性

再接続情報 94

更新

システム・テーブル 9

システム・テーブルの直接更新 9

システム・プロシージャ 9

構造体

設定 33

さ

- サイズ
 - ロー 98
 - 作業領域
 - 削除 115
 - 削除
 - 作業領域 115
 - 削除されたロー数 98
 - 参照情報
 - dbcc** テーブル 113
 - システム・テーブル 3
 - 参照整合性制約
 - sysconstraints* テーブル 31
 - sysobjects* テーブル 65–69
 - sysreferences* テーブル 86
- し
- シーケンス・ツリー、オブジェクト 76
 - 時間範囲
 - systemranges* システム・テーブル 102
 - システム・テーブル 1–10
 - allow updates to system tables** パラメータ 9
 - master* データベース 1–2
 - キー 52
 - 更新 9
 - 直接更新 9
 - トリガ 9
 - パーミッション 7
 - システム標準の役割
 - sysloginroles* テーブル 57
 - sysserverroles* テーブル 96
 - システム・プロシージャ
 - 更新 9
 - 情報 (サーバ)
 - 設定パラメータ 29, 33
 - データベース 36

す

- スクラッチ・データベース 12
- ストアド・プロシージャ
 - オブジェクトの依存性 40
 - システム・テーブル・エントリ 27, 65–69, 76
- スライス
 - システム・テーブル・エントリ 95
- スレッシュホールド
 - sys thresholds* テーブル 100

せ

- 制約
 - sysconstraints* テーブル 31
 - sysreferences* テーブル 86
 - システム・テーブル・エントリ 27, 65
- セグメント
 - syslices* テーブル 95
 - syssegments* テーブル 91
 - システム・テーブル・エントリ 91
- 設定パラメータ
 - システム・テーブル 29, 33

そ

- ソート順
 - syscharsets* システム・テーブル 23

た

- ダンプ・デバイス
 - sysdevices* テーブル 41
 - システム・テーブル・エントリ 41

ち

- 直接更新
 - システム・テーブル 9

つ

月の値

- syslanguages* テーブル 53
- 代替言語 53
- 短縮形 (しょうりゃくけい) 53

て

ディスク・デバイス

- sysdevices* テーブル 41
- ディスク・ミラーリング
- sysdevices* テーブルの status 42
- ディスク割り付けの集まり (disk allocation pieces) 108

データ型

- リスト 105
- systypes* テーブル 105–107
- 階層 105

データベース

- システム・テーブル・エントリ 36

データベース・オブジェクト

- sysobjects* テーブル 65–69
- 依存性 40

データベース・デバイス

- sysdevices* テーブル 41
- システム・テーブル・エントリ 41

テープ・ダンプ・デバイス

- sysdevices* テーブル 41

テーブル

- monCachedObject** 132
- monCachePool** 133
- monCIPC** 138
- monCIPCEndpoints** 139
- monCIPCLinks** 140
- monCIPMesh** 141
- monCLMObjectActivity** 143
- monClusterCacheManager** 145
- monCMSFailover** 146
- monDataCache>** 149
- monDataCacher** 147
- monDBRecoveryLRTypes** 151
- monDeadlLock** 152
- monDeviceIO** 155
- monEngine** 157

- monErrorLog** 159, 160
- monIOController** 162
- monLicense** 164
- monLocks** 165
- monLogicalCluster** 172
- monLogicalClusterAction** 174
- monLogicalClusterInstance** 175
- monLogicalClusterRoute** 176
- monNetworkIO** 177
- monOpenDatabases** 178
- monOpenObjectActivity** 180
- monOpenPartitionActivity** 185
- monPCIBridge** 191
- monPCIBridge>** 190
- monPCISlots** 192
- monPCM** 193
- monProcedureCache** 195
- monProcedureCacheMemoryUsage** 196
- monProcedureCacheModuleUsage** 197
- monProcess** 198
- monProcessActivity** 200
- monProcessLookup** 202, 203
- monProcessNetIO** 204
- monProcessObject** 205
- monProcessProcedures** 206
- monProcessSQLText** 208
- monProcessStatement** 209
- monProcessWaits** 210
- monProcessWorkerThread** 211
- monState** 220
- monStatementCache** 221
- monSysLoad** 222
- monSysPlanText** 223
- monSysSQLText** 224
- monSysStatement** 225
- monSysWaits** 227
- monSysWorkerThread** 228
- monTableColumns** 229
- monTableCompression** 231
- monTableParameters** 232
- monTables** 233
- monTableTransfer** 234
- monTempdbActivity** 236
- monWaitClassInfo** 239
- monWaitEventInfo** 240
- monWorkload** 241

monWorkloadPreview 242
monWorkloadProfile 243
monWorkloadRaw 244
monWorkQueue 245
monCachedProcedures テーブル 134
 オブジェクトの依存性 40
 システム・テーブル・エントリ 24, 65
monIOQueue 163
 デバイス
 システム・テーブル・エントリ 41
 デフォルト
 システム・テーブル・エントリ 27, 65–69, 76
 転送されたロー
 数 98

と

統計
 システム・テーブル 97, 98
 トランザクション 103
 トランザクション・ログ
 システム・テーブル・エントリ 65–69
 トリガ
 オブジェクトの依存性 40
 システム・テーブル 9
 システム・テーブル・エントリ 27, 65–69, 76

な

名前
 ソート順 23
 文字セット 23
 名前付き時間範囲
 systemranges システム・テーブル 102

に

偽のテーブルのマテリアライズ 6

は

パーミッション
 sysprotects テーブル 80
 システム・テーブル 7
 システム・テーブル・エントリ 80

ひ

sysdevices 内の *status* 42
 ビュー
 オブジェクトの依存性 40
 システム・テーブル・エントリ 24, 27, 65–69, 76

ふ

プライマリ・キー
 syskeys テーブル 52
 プラン
 オブジェクト 76
 プロセス (サーバのタスク)
 sysprocesses テーブル 77
 システム・テーブル・エントリ 77
 プロセスのブロック 77
 分散トランザクション管理 (DTM) 32

へ

ページ、グローバル・アロケーション・マップ 46
 ページ、データ
 数 98

ま

マッピング
 sysusages テーブル 108

め

メッセージ

sysmessages テーブル 63*sysusermessages* テーブル 109

ユーザ定義 109

も

モニタリング・テーブル 131

説明とリスト 131

や

役割

sysroles テーブル 89*sysserverroles* テーブル 96

役割、システム

sysloginroles テーブル 57

ゆ

ユーザ

syslogins テーブル 58–60*sysusers* テーブル 110

システム・テーブル・エントリ 58–60, 110

ユーザ定義の役割

sysserverroles テーブル 96

予約済みカラム 8

り

リスト

システム・テーブル 1–4

リストと説明、モニタリング・テーブル 131

リソース制限

sysresourcelimits テーブル 88

リモート・サーバ

sysserver テーブル 92

システム・テーブル・エントリ 92

リモート・プロシージャ・コール

sysremotelogins テーブル 87*sysserver* テーブル 92

リモート・ログイン

sysremotelogins テーブル 87

システム・テーブル・エントリ 87

領域の割り付け

sysusages テーブル 108

システム・テーブル・エントリ 108

る

ループ

syslogs の変更と無限ループ 61

ルール

システム・テーブル・エントリ 27, 65–69, 76

ろ

ロー、インデックス

サイズ 98

リーフのサイズ 98

ロー、データ

数 98

サイズ 98

ロー、テーブル

サイズ 98

ログイン

syslogins テーブル 58–60*sysremotelogins* テーブル 87

“probe” 60

ロック

システム・テーブル・エントリ 55

わ

“probe” ログイン・アカウント 60

ESP。「拡張ストアド・プロシージャ」参照