



用語解説

Adaptive Server[®] Enterprise

15.7

ドキュメント ID : DC36502-01-1570-01

改訂 : 2011 年 9 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、**Sybase trademarks** ページ (<http://www.sybase.com/detail?id=1011207>) で確認できます。Sybase およびこのリストに掲載されている商標は、米国法人 Sybase, Inc. の商標です。® は、米国における登録商標であることを示します。

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 Sun Microsystems, Inc. の商標または登録商標です。

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

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

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

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

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

用語解説

この用語解説では、Adaptive Server® のマニュアルで使用されている用語について説明します。

以下の文字を選択すると、『用語解説』内のその文字の項に移動できます。

記号	数字	A	B	C	D	E	F	G	I	J	L	M	N	O	P
Q	R	S	T	U	X	あ	い	う	え	お	か	き	く	け	こ
さ	し	す	せ	そ	た	ち	て	と	な	に	ね	の	は	ひ	ふ
へ	ほ	ま	み	む	め	も	や	ゆ	よ	ら	り	る	れ	ろ	わ

記号

“dbo” アカウント (“dbo” account)

「データベース所有者 (Database Owner)」参照。

“sa” ログイン・アカウント (“sa” login account)

「システム管理者 (System Administrator)」参照。

数字

16 進文字列 (hexadecimal string)

16 進コード化バイナリ文字列。プレフィクス 0x で始まり、数字の 0 から 9 およびアルファベットの大文字か小文字の A から F までで構成されます。16 進文字列の解釈の方法はプラットフォームによって異なります。プレフィクスの次の最初のバイトが一番重要になるシステムもあれば、最後のバイトが重要なシステムもあります。たとえば、システムによっては 0x0100 が 1 と変換され、他のシステムでは 256 と変換されます。

A

Adaptive Server

Sybase クライアント/サーバ・アーキテクチャのサーバ (11.5 より前のバージョンでは SQL Server と呼ばれる)。Adaptive Server は、複数のデータベースおよび複数のユーザの管理、ディスク上のデータの実際のロケーションの監視、論理データ記述から物理データ記憶領域へのマッピングの管理、およびメモリ内のデータ・キャッシュとプロシージャ・キャッシュの保守を行います。

Adaptive Server エンジン (Adaptive Server engine)

「エンジン (engine)」参照。

Adaptive Server ログイン (Adaptive Server login)

ユーザが Adaptive Server へのログインに使用する名前。Adaptive Server のシステム・テーブル syslogins にそのユーザ用のエントリが存在すれば、ログインが有効になります。

API (application program interface)	「アプリケーション・プログラム・インタフェース (application program interface: API)」参照。
B	
B ツリー (B-tree)	平衡木またはバイナリ・ツリーの省略形。Adaptive Server は、B ツリーのインデックスを使用します。B ツリーのリーフ・ページはどれも、インデックスのルート・ページから等距離に配置されます。B ツリーは、一貫性のある予測可能なパフォーマンス、すぐれた逐次およびランダム・レコード検索能力、フラットなツリー構造を提供します。
Backup Server	Adaptive Server の代わりに、選択されたデータベースとトランザクション・ログに対してローカルまたはリモートのバックアップ(ダンプ)とリストア(ロード)を行います。Backup Server は Adaptive Server と同じシステム上で稼働している必要があります。
bcp	「バルク・コピー (bulk copy)」参照。
Bean	プログラムのビジネス・ロジックを含むサーバ側のコンポーネント。実行時に、アプリケーション・クライアントは Enterprise Bean の関数を呼び出して、このビジネス・ロジックを実行します。
C	
CA 証明書 (CA certificate)	信頼されたルート証明書ともいいます。CA 証明書は、起動時にサーバがロードする信頼された CA のリストです。
certauth	サーバ証明書の要求を CA の署名付き証明書に変換する Sybase ユーティリティ。
certpk12	サード・パーティの PKCS12 フォーマットの証明書を、Adaptive Server と Open Client/Open Server が認識できるフォーマットに変換する Sybase ユーティリティ。
certreq	パブリック・キーおよびプライベート・キーのペアと、証明書の要求を生成する Sybase ユーティリティ。
CipherSuite	SSL が有効なアプリケーションで使用されるキー交換アルゴリズム、ハッシュ・メソッド、暗号化メソッドの優先リスト。
CIS	「コンポーネント統合サービス (Component Integration Services : CIS)」参照。
CORBA	「CORBA (Common Object Request Broker Architecture)」参照。

CORBA (Common Object Request Broker Architecture)	Common Object Request Broker Architecture。OMG (Object Management Group) によって定義されるオープン分散オブジェクト・コンピューティングの基礎構造。一般的なネットワーク・プログラミング・タスクの多くを自動化します。アーキテクチャのキー・コンポーネントである ORB (Object Request Broker) は、オブジェクト間でクライアント/サーバ関係を確立するミドルウェアです。これによって、クライアントは同一マシンまたはネットワーク上のマシンのサーバ・オブジェクトでメソッドを呼び出せるようになります。
CPU コスト (CPU cost)	クエリを処理するためにオプティマイザが必要とする CPU の概算値。
D	
DAC	「 任意アクセス制御 (discretionary access controls: DAC) 」 参照。
datachange	テーブルのカラムに対して行われる更新、挿入、削除の数の単位。
DataWindow オブジェクト (DataWindow object)	「 クエリ・オブジェクト (query object) 」 参照。
dbcc コマンド (dbcc commands)	データベース一貫性チェッカ (dbcc) に対する命令。データベースの論理的一貫性と物理的一貫性を検査します。
DDL	「 データ定義言語 (data definition language : DDL) 」 参照。
default 句 (default clause)	create table 文内でカラムにデフォルト値を指定します。
deleted テーブル (deleted table)	delete 文の削除された値を持つロー、または update 文の更新前の値 (更新前のイメージ) を格納している擬似テーブルです。
deterministic プロパティ (deterministic property)	計算カラムのプロパティの 1 つ。指定した入力のセットから常に同じ結果を返すように指定します。
DIT	「 ディレクトリ情報ツリー (directory information tree: DIT) 」 参照。
DLL	「 ダイナミック・リンク・ライブラリ (dynamic link library: DLL) 」 参照。
DSS	大量のデータを処理するクエリの実行を特徴とするアプリケーション。
DTD	「 ドキュメント・タイプ定義 (document type definition: DTD) 」 参照。
DTX パティシパント (DTX participant)	DTX パティシパント (Distributed transaction participant) は、 Adaptive Server が、従属する Adaptive Server へ送信されたトランザクションの調整に使用する内部メモリ構造です。

E

EAServer Adapter EAServer にクライアント・インタフェースを提供するアダプタ。EAServer Adapter により、ビジネス・イベント処理の一部として Enterprise JaveBean メソッドを実行できるようになります。

Enterprise Application Server EAServer (Jaguar と呼ばれる)。Web アプリケーションおよびコンポーネント・ベース・アプリケーション用のスケーラブルで安全なトランザクション認識プラットフォームを持つ統合開発と配備環境。SEEB (Sybase Enterprise Event Broker) は、EAServer のサービスとして実行されます。

F

for load テープからリストアするためにデータベースの作成を指定します。

for トリガ (for trigger) Adaptive Server が現在サポートしているトリガ。このトリガはテーブルの insert/update/delete 文の後で起動するので、after トリガと呼ばれることもあります。

G

guest model データベースの sysusers テーブル内のユーザ名。有効な Adaptive Server ログインを持つユーザは、このユーザ名で model から作成されたデータベースを利用することができます。権限に制限があります。

I

I/O 「デバイス I/O (device I/O)」参照。

IDENTITY カラム (IDENTITY column) テーブルの各ローをユニークに識別するシステム生成値が格納される **カラム (column)**。IDENTITY カラムには、Adaptive Server が自動的に生成するユニークな番号 (送り状番号や従業員番号など) が保存されます。IDENTITY カラムの値は、テーブル内の各ローをユニークに識別します。

image 関数 (image function) 「テキストおよびイメージ関数 (text and image function)」参照。

insert table insert 文の挿入された値を持つロー、または update 文の更新後の値 (更新後のイメージ) を格納している擬似テーブルです。

instead of トリガ (instead of trigger) ビューに定義され、update/insert/delete 文の標準的なアクションを置き換えるトリガ。

int 符号付き 32 ビット整数値。

interfaces ファイル (interfaces file) Adaptive Server へ接続された各マシンに必要なオペレーティング・システム・ファイル。デフォルトでは、このファイルは SYBASE 環境変数で指定されるディレクトリにあります。ファイル名はまちまちですが、通常 UNIX システムでは *interfaces*、PC では *sql.ini* です。

interfaces ファイルの各エントリによって、ホスト・マシンが Adaptive Server に接続する方法が決まります。interfaces ファイルのエントリには、インストール環境の名前と Adaptive Server が提供するサービスのリストが含まれます。

isql [Adaptive Server](#) へのコマンドライン・インタフェース。

J

Java アプリケーション (Java application)

Java 用語では、アプリケーションとはホスト・サーバやブラウザを稼働させる必要のない Java プログラムのこと。ユーザはウィンドウやメニューを表示して、コントロールと対話します。アプリケーションは、中間層サーバやデータベース・サーバと接続できます。

L

lava 演算子 (lava operator)

基本の演算の 1 つを実装し、[オプティマイザ \(optimizer\)](#) が lava クエリ・プランの一部として選択できる、独立したソフトウェア・オブジェクト。lava 演算子には、データベース・[テーブル \(table\)](#) から [ロー \(row\)](#) を読み取る ScanOp、マージ・ジョインを実装する MergeJoinOp、テーブルにローを挿入する InsertOp などがあります。lava 演算子は 32 種類あります。「[反復子 \(iterator\)](#)」参照。

lava クエリ実行エンジン (lava query execution engine)

オプティマイザによって選択された lava クエリ・プランを実行する Adaptive Server のモジュール。クエリ・プランは、プラン内の最上位の演算子 (RootOp) のメソッドを呼び出すことによって実行されます。最上位の演算子は子演算子のメソッドを呼び出し、子演算子がさらにその子演算子のメソッドを呼び出すというようにしてリーフ演算子までのメソッドが呼び出され、必要に応じて結果ローが生成されます。結果ローはリーフ演算子ノードで生成され、演算子ツリーを経由して最上位の RootOp に返されます。RootOp は、結果ローをコンシューム (クライアントに送信したり、変数に値を割り当てるなど) します。

lava クエリ・プラン (lava query plan)

複数の lava 演算子からなる「逆」ツリー構造。最上位の演算子が 1 つまたは複数の子演算子を持ち、子演算子がさらに 1 つまたは複数の子演算子を持つというようにして、演算子の逆ツリー構造が構成されます。オプティマイザが lava クエリ・プランのツリーの形と演算子を選択し、lava クエリ実行エンジンがプランを実行します。

LDAP (Lightweight Directory Access Protocol)

業界標準のディレクトリ・サービスへのアクセス・メソッド。コンポーネント (アプリケーション、ほかのサーバなど) は DN (識別名) によって、LDAP サーバから情報を検索することができます。LDAP サーバは、企業やネットワーク全体で使用されるサーバ、ユーザ、ソフトウェア情報を格納および管理します。LDAP では通信プロトコル、およびクライアント/サーバ間で交換されるメッセージのコンテンツを定義します。

lightweight directory access protocol

「[LDAP \(Lightweight Directory Access Protocol\)](#)」参照。

LIO

「[論理 I/O コスト \(logical I/O cost\)](#)」参照。

- LOB** ラージ・オブジェクト。LOB は、`text`、`image`、および `unitext` データ型を指します。LOB 値は、非常に大規模なことがあるため、データベースまたはメモリへの保存時に特別な考慮が必要な場合があります。
- LRU (最も長い間使用されていない) バッファ再使用方式 (least recently used (LRU) buffer reuse strategy)** データ・キャッシュ内の最も長い間使用されていないバッファを置き換えるキャッシュ方式。クリーンなデータ・ページがデータ・キャッシュ上の LRU 側の終端から取り出され、ディスクから読み込まれたページが保存されます。データ・キャッシュのページ・チェーンの MRU (最も最近に使用された) 側の終端に新しいページが配置され、メモリに残ります。
- M**
- MAPI** 「[MAPI \(Messaging Application Programming Interface\)](#)」 参照。
- MAPI (Messaging Application Programming Interface)** Microsoft によって開発された電子メールアプリケーション・プログラミング・インタフェース。[Adaptive Server](#) は、データの送受信用インタフェースを従来のクライアント/サーバ接続ベースの機能ではなく電子メールによって提供するために、MAPI を利用しています。
- master データベース (master database)** 全体として、ユーザ・データベースと [Adaptive Server](#) のオペレーションを制御するシステム・データベース。`master` と呼ばれ、ユーザ・アカウント、実行中の処理、システム・エラー・メッセージなどを追跡します。
- MenuBar オブジェクト (MenuBar object)** [PowerJ](#) を使用した [Java アプリケーション \(Java application\)](#) の構築において、フォームに表示されるすべてのメニューを表します。MenuBar オブジェクトは、Frame クラスに基づくフォームに対してのみ追加できます。
- MIME** 「[MIME \(Multipurpose Internet Mail Extensions\)](#)」 参照。
- MIME (Multipurpose Internet Mail Extensions)** 多目的インターネット・メール拡張機能。インターネットで複雑なメッセージをやりとりするためのフォーマット。Web ブラウザとサーバは、MIME タイプを使ってサーバからブラウザに転送されるメッセージを記述します。
- model データベース (model database)** 新しいユーザ・データベースのためのテンプレート。[Adaptive Server](#) をインストールすると、インストール処理によって `model` が作成されます。`create database` を実行するたびに、[Adaptive Server](#) は `model` のコピーを作成し、場合によっては要求されたサイズまで拡張します。
- MRU (最も最近に使用された) 置換方式 (most recently used (MRU) replacement strategy)** テーブル・スキャンとノンクラスタード・インデックス・スキャンのためのキャッシュ方式。最適化は、特定のクエリで一度だけアクセスする必要のあるページを決定するときにこの方式を使用します。MRU/LRU チェーンにすべてのページを追加する代わりに、ページはクエリで検査が終了するとすぐにフラッシュされ、クエリに必要な次のページがバッファに読み込まれます。
- N**
- NULL** 明示的に値の割り当てられていないパラメータやオプションに与えられる値。NULL は 0 でも空白でもありません。NULL の値は、他の値と比べて大きいとも、小さいとも、同じであるともみなされません。他の NULL との比較においても同様です。

O

OAM 「オブジェクト・アロケーション・マップ (object allocation map: OAM)」 参照。

ODBC 「ODBC (Open Database Connectivity)」 参照。

ODBC (Open Database Connectivity) Microsoft によって定義された ODBC インタフェースは、Windows や Windows NT 環境でのデータベース管理システムの標準インタフェースです。

OLTP 「オンライン・トランザクション処理 (online transaction processing: OLTP)」 参照。

OR 方式 (OR strategy) or を使うクエリと in (値リスト) を使うクエリを処理するための **オプティマイザ (optimizer)** 方式。テーブルからのデータ・ローの検索や限定には、インデックスを使用します。ロー ID はワーク・テーブルに保存されます。すべてのローが検索がされると、ワーク・テーブルは重複を取り除くためにソートされ、ロー ID がテーブルのデータ検索に使用されます。

open 論理クラスタ (open logical cluster) (クラスタの概念) 定義済みのルートのない接続を受け入れる論理クラスタ。デフォルトでは、システムの論理クラスタにオープン・プロパティがありますが、別の論理クラスタにオープン・プロパティを設定できます。オープン・プロパティを持つことのできる論理クラスタは一度に 1 つだけです。

P

PIO 「物理 I/O コスト (physical I/O cost)」 参照。

PowerJ **Adaptive Server** で Web 用のアプリケーションを開発するために使用する Java コンポーネント。

public データベースに登録されたすべてのユーザは、グループ “public” のメンバです。この権限レベルのユーザはテンポラリ・テーブルを作成でき、オブジェクトの所有者が “public” に対してパーミッションを与えたオブジェクトにアクセスできます。

Q

QP (クエリ処理) 測定基準 (QP (query processing) metrics) クエリ処理 (QP) 測定基準は、クエリの実行において経験的な測定基準値を識別して比較します。クエリが実行されると、この測定基準は、QP 測定基準での比較の基礎となる定義済みの測定基準のセットと関連付けられます。

R

RDBMS 「リレーショナル・データベース・マネジメント・システム (Relational Database Management System: RDBMS)」 参照。

RID 「ロー ID (row ID: RID)」 参照。

RPC 「リモート・プロシージャ・コール (remote procedure call: RPC)」 参照。

runserver ファイル (runserver file)	Adaptive Server またはバックアップ・サーバを再起動するときに、参照として使用するファイル。デフォルトでは、runserver ファイル名は <i>RUN_servername</i> で、Adaptive Server のインストール時に作成されます。Runserver ファイルは、 <i>\$\$SYBASE/install</i> ディレクトリに作成されます。
S	
SARG	「 探索指数 (search argument) 」参照。
SEEB	「 Sybase Enterprise Event Broker (SEEB) 」参照。
select リスト (select list)	select 文のメイン句に指定する カラム (column) 。 従属 (dependent) ビューでは、従属ビューを有効にするために、基本となるすべてのビューにターゲット・リストを持たせる必要があります。
SMP	「 対称型マルチプロセッシング (symmetric multiprocessing: SMP) 」参照。
SQL	「 構造化問合せ言語 (Structured Query Language: SQL) 」参照。
SQL Server	「 Adaptive Server 」参照。
SQL 抽出テーブル (SQL derived table)	クエリ式の評価によって 1 つ以上のテーブルに基づいて定義される テーブル (table) 。SQL 抽出テーブルは定義されたクエリ式の中で使用され、クエリの実行中のみ存在します。「 抽出テーブル (derived table) 」と「 抽象プランの抽出テーブル (abstract plan derived table) 」参照。
sql.ini ファイル (sql.ini file)	ワークステーションから接続可能な各 Adaptive Server の定義を記述した interfaces ファイル。このファイルは Adaptive Server に接続する各クライアント・マシン上に格納する必要があります。sql.ini ファイルの各エントリには、クライアント・マシンまたはホスト・マシンを Adaptive Server に接続する方法が示されています。このファイルには、Adaptive Server の名前、Adaptive Server で提供されるサービスのリスト、各サービスへ接続するために使用するポートが記述されています。
SSL	「 SSL (Secure Sockets Layer) 」参照。
SSL (Secure Sockets Layer)	SSL は、クレジットカード番号、株式売買、銀行取引などの機密情報を、インターネット上で安全に転送するための標準です。
SSL ハンドシェイク (secure sockets layer handshake)	サーバとクライアント間の一連の往復 I/O。サーバがサーバ証明書を提示し、サーバとクライアントがネゴシエートしてセキュリティ・メカニズムについて合意が得られないかぎり、データは送信されません。
SSL フィルタ (secure sockets layer filter)	Adaptive Server の interfaces ファイルの master および query 行に追加されたフィルタ。特定ポートへのすべての接続は、SSL プロトコルをサポートしないかぎり接続が確立されないようにします。
Sybase Enterprise Event Broker (SEEB)	Enterprise Event Broker は、Sybase サーバでメッセージの送受信を可能にする基礎構造を提供します。

T

TCP/IP クライアント／サーバ・コンピューティングで 사용되는プライマリ・トランスポート・プロトコル。TCP/IP は、インターネット上のデータ転送を制御するプロトコルです。

TPS 1 秒当たりのトランザクション数。「スループット (throughput)」参照。

Transact-SQL Sybase [Adaptive Server](#) で使用される SQL 構文。

trusted ユーザ (trusted user) セッション (session) 内の一定の sensitivity レベルの範囲で操作を認められているユーザ。ユーザは操作をする範囲レベルを指定してログインします。ユーザは現在の読み込みレベルや書き込みレベルをセッション中に変更できます。

trusted ルート・ファイル (trusted root file) サーバ起動時にロードされる trusted 認証機関 (CA) のリスト。

U

Unicode 日本語、中国語、ロシア語、フランス語、ドイツ語など、650 以上の言語をサポートする国際的な文字セット。Unicode を使用すると、プラットフォームに関係なく、同じサーバ内で多数の言語グループのさまざまな言語を一緒に使用できます。

UTF-16 各 UCS-2 コード値はそれ自体を表します。BMP (Basic Multilingual Plane: 0..0xFFFF) の範囲外のコード値は、サロゲート・ペアという特別な 16 ビットのコードを使用して表現します。これにより、 2^{20} バイト (1MB) のコード値を追加で表現できますが、そのためには 4 バイトを使用する必要があります。

UTF-8 [Unicode](#) 標準の可変長コード化形式で、8 ビット・シーケンスを使用します。上位ビットは、バイトがシーケンスのどの部分に属するかを示します。

X

XP Server [Adaptive Server](#) の代わりに拡張ストア・プロシージャ (stored procedure) を実行する Open Server™ プロセス。XP Server と [Adaptive Server](#) は同じマシンで実行し、リモート・プロシージャ・コール (remote procedure call: RPC) を通じて相互に通信します。

あ

アカウント・ロック (account locking) ユーザが [Adaptive Server](#) にログインできないようにする [Adaptive Server](#) の機能。ただし、ユーザのアカウントが、データベースやオブジェクトを所有することは許可します。ユーザが他のユーザに付与したパーミッションや、ユーザが所有するデータベース・オブジェクトの使用を妨げることなく、ユーザのアクセスを拒否できます。ログイン・ロックとも呼ばれます。

空き領域スレッシュホールド (free-space threshold) ユーザ指定のスレッシュホールド。セグメント上の領域と、セグメントで利用できる領域が指定領域を下回ったときに実行される動作を指定します。

アクション (action)	(クラスタの概念) 計画済みのフェールオーバ、ダウン時間、またはその他の管理タスクを開始するために指定された時間に 1 つ以上のインスタンスを停止するユーザ実行コマンド。アクションによって、インスタンスの状態が変化します。
アクセス (access)	テーブルやビューにおける <code>select</code> コマンド、 <code>insert</code> コマンド、 <code>update</code> コマンド、 <code>delete</code> コマンドの使用。
アクセス・メソッド (access method)	クエリの結果を導出するために必要なデータを探すための方法。アクセス・メソッドには、逐次メソッドと並列メソッドがあります。逐次アクセス・メソッドには、 テーブル・スキャン (table scan) 、 ノンクラスタード・インデックス (nonclustered index) 、 クラスタード・インデックス (clustered index) があります。
アダプタ (adapter)	SEEB (Sybase® Enterprise Event Broker) のコンポーネント。SEEB と外部アプリケーションまたはメッセージング・システムの間で、インタフェースを提供します。アダプタは、イベントを検出してその内容を検証し、それらをインフロー・プロセッサに渡します。また、アウトフロー・プロセッサで SEEB イベントを受け取り、それを外部アプリケーションへエクスポートします。
アドホック文 (ad hoc statement)	ストアド・プロシージャに含まれず、動的 SQL でもない SQL 文。Transact-SQL バッチはアドホック SQL 文で構成されます。
アドレス・ロック (address lock)	ネットワーク・バッファや内部構造体など、特定のリソースに適用されるロックの種類。アドレス・ロックは、インデックス・ページをロックする場合にも使われます。
アプリケーション (application)	Adaptive Server と対話するクライアント・プログラム。特に、テーブル・エントリ <code>sysprocesses</code> 内の <code>program_name</code> カラムのアプリケーション名のこと。クライアント・ライブラリは、アプリケーションの <code>program_name</code> を指定するメカニズムを提供します。
アプリケーション・プログラム・インタフェース (application program interface: API)	クライアント・アプリケーションと Adaptive Server とのインタフェースの役割を果たすサブルーチン。 ライブラリ (libraries) ともいいます。
アロケーション・ページ (allocation page)	アロケーション・ユニット (allocation unit) の最初のページ。アロケーション・ユニット内のすべてのページの使用状況を追跡します。
アロケーション・ユニット (allocation unit)	1/2 MB、つまり 256 × 2K ページの論理単位。 <code>disk init</code> コマンドは Adaptive Server 用に新しいデータベース・ファイルを初期化し、それをアロケーション・ユニットに分割します。
暗号化 (encryption)	意図された受信者以外のユーザから情報を保護するため、暗号化シグニチャを使用して情報をコード化するプロセス。
暗黙の変換 (implicit conversions)	データ型 (datatype) を比較するために、 Adaptive Server が自動的に行うデータ型の変換。

い

異機種サーバ
(heterogeneous
server)

Informix、DB2 データベースなど、[Adaptive Server](#) 環境ではないリモート・サーバ。

異機種データ
(heterogeneous
data)

Oracle、Informix、DB2 など、Sybase 以外のデータ・ソースのデータ。

移行モード
(transitional mode)

フェールバック移行モードは、[Adaptive Server](#) がフェールオーバー・モードからノーマル・コンパニオン・モードに移行するときに発生するモード。フェールバック移行モードは、一般的に持続時間が非常に短いものです。

意思決定支援システム
(decision support
system: DSS)

大量データを処理するクエリをサポートするデータベース・システム。通常は、業務決定のために使用します。DSS クエリは、一般にはテーブル全体または、テーブルの大多数にアクセスし、多くのテーブルにジョインを使用し、大量の結果セットの要約データを返します。DSS アプリケーションはスケジュールに従って実行されることが多く、たとえば、毎日の業務終了のときに定型のレポートを作成するなどの用途に使用します。「[データ・ウェアハウス \(data warehouse\)](#)」参照。

依存性 (dependency)

あるオブジェクトが別のオブジェクトを参照するときに発生するオブジェクト間の関係。たとえば、ストアド・プロシージャがテーブルを参照する場合などです。

一意性制約 (unique
constraint)

指定した[カラム \(column\)](#) 内の非 null 値がユニークであることを要求する[制約 \(constraint\)](#)。テーブル ([table](#)) 内の指定カラムに同じ[ロー \(row\)](#) があることは認められません。unique 制約は指定したカラムにユニークな[インデックス \(index\)](#) を作成し、このデータの整合性を保ちます。

位置付け更新
(positioned update)

where current of 句を使用して、カーソルで実行された更新。「[検索更新 \(searched update\)](#)」参照。

位置付け削除
(positioned delete)

where current of 句を使用して、カーソルを使って実行された delete。「[検索削除 \(searched delete\)](#)」参照。

位置番号 (position
number)

選択したリスト内のカラムの位置。位置番号はカラム名の代わりに使用できません。

意図的ロック
(intent lock)

テーブル・レベルのロックの 1 つ。共有[意図的ロック \(intent lock\)](#) は、ページまたはその[テーブル \(table\)](#) の [ロー \(row\)](#) に共有ロックがあることを表します。これによって、他のトランザクションは排他的テーブル・レベル・ロックを実行できなくなります。排他意図的ロックは、そのテーブルのページに排他ロックや更新ロックがあることを表します。これによって、他のトランザクションは排他的テーブル・レベル・ロックや共有テーブル・レベル・ロックを実行できなくなります。

イベント (event)	Sybase Enterprise Event Broker (SEEB) 内では、イベントとは、インポートされ、プロセッサ間を通して外部データベースへエクスポートされるオブジェクト。SEEB 外では、メッセージまたはデータベース・オペレーションの場合もあります。
イベントの内容 (event content)	イベント・スキーマとイベント・コンテナの組み合わせ。
イベント・ブローカ (event broker)	バス、キュー、または同等の送信先にイベントをルート指定、変換、配信するアプリケーション。
イベント・プロセッサ (event processor)	ルートを介してイベントを送信し、ルートに従ってイベントを処理する Sybase Enterprise Event Broker (SEEB) コンポーネント。ルートは2つ以上のイベント・プロセッサのシーケンスで構成されます。イベント・プロセッサは、イベント・プロセッサ論理に従ってイベントに作用します。SEEB イベント・プロセッサには、インフロー・イベント・プロセッサ、アウトフロー・イベント・プロセッサ、転送プロセッサ、ブランチ・プロセッサ、変換プロセッサなどがあります。
インオーダー・ジョイン (in-order join)	外部ジョイン (outer join) のジョイン属性の一部 (または全部) が、ソートまたはインデックス・スキャンを実行した場合のように順番に並んでいるジョイン操作。
インスタンス (instance)	(クラスタの概念) 共有ディスク・クラスタにある Adaptive Server。
インスタンスのステータス (instance state)	(クラスタの概念) 論理クラスタ内のインスタンスの論理クラスタによって検知されるステータス。つまり、物理的にオンラインであるインスタンスが、特定の論理クラスタにオフラインとして検知される場合もあります。
インスタンスのプール (instance pooling)	単一のコンポーネント・インスタンスで複数のクライアントにサービスを提供することを可能にします。インスタンスのプールによって、コンポーネント・インスタンスを繰り返し割り付けることによるリソースの消耗を防止します。
インスタンス番号 (instance number)	(クラスタの概念) Adaptive Server 共有ディスク・クラスタ内の名前付きインスタンスをユニークに識別する番号。
インデックス (index)	データ・テーブルからのキー値と、その値が含まれたページに対するポインタから構成される データベース・オブジェクト (database object) 。インデックスは、 Adaptive Server にディスク上のテーブル・カラムのデータの位置を伝え、データ・ローへ高速にアクセスします。
インデックス・カバリング (index covering)	ノンクラスタード・インデックスのリーフ・レベルのページに、クエリの条件を満たすのに必要なデータが含まれているデータ・アクセス条件。インデックスには、クエリ句のカラムだけでなく、場合によっては、select リスト中のすべてのカラムを含む必要があります。サーバはインデックスの リーフ・レベル (leaf level) だけでクエリの条件を満たすことができます。インデックスがクエリの条件を満たす場合、サーバはデータ・ページをアクセスしません。

インデックス共用体 (index union)	1つのテーブルの、2つ以上のインデックスで取得した複数のロー ID (row ID: RID) を、重複ローの削除を使用して結合するアクセス・パス。“OR”で結合された複数の SARG を使用するテーブル・スキャンの結果セットの条件を満たす RID のセットを取得します。
インデックス交差 (index intersection)	1つのテーブルの、2つ以上のインデックスで取得した複数の RID を結合するアクセス・パス。“AND”で結合された複数の SARG を使用するテーブル・スキャンの結果セットの条件を満たす RID のセットを取得します。
インデックスの選択性 (index selectivity)	インデックス (index) 内の重複キーの値の比率。ユニークなキーの検索など、オプティマイザ (optimizer) で1ローだけを選別できる場合、インデックスは選択性が低いといえます。ユニークでないエントリに対するインデックスは選択性が低いといえます。“M”または“F”(男性または女性)などの値に対するインデックスは、選択性が非常に低いといえます。
インデックス・ページ・クラスタ率 (index page cluster ratio)	追加のエクステント I/O を必要としない、ページ・チェーン経由のインデックス・リーフ・ページ・アクセスの率。インデックス・ページ・クラスタ率は、データオンリー・ロック・テーブル上のクラスタード・インデックス (clustered index)、およびノンクラスタード・インデックス (nonclustered index) に対して管理されます。
インメモリ・データベース (in-memory database)	データまたはログの保存にディスク記憶領域を使用せず、すべてが名前付きキャッシュに保存されたデータベース。インメモリ・データベースの持続性レベルは no_recovery です (Adaptive Server の再起動時にデータベースが再作成されます。) インメモリ・データベースでは、すべてがメモリに常駐し、持続性が低いことを利用して、ディスク常駐のデータベースよりも高いパフォーマンスを提供するとともに、高いトランザクション・スループットを管理できます。
う	
ウォッシュ・エリア (wash area)	MRU/LRU ページ・チェーンの LRU 側の終端付近のバッファ・プール (buffer pool) の領域。ページがウォッシュ・エリアに入ると、Adaptive Server はページに対する非同期書き込みを開始します。ウォッシュ・エリアは、ディスク I/O の実行が必要なクエリに対し、LRU でクリーンなバッファを提供します。
ウォッシュ・マーカ (wash marker)	MRU/LRU チェーン上にあるキャッシュ内のポイント。Adaptive Server データ・キャッシュは、バッファの MRU/LRU (最も最近に使用された/最も長い間使用されていない) チェーンとして管理されます。キャッシュ内のバッファが古くなるにつれ、バッファは MRU 側の終端から LRU 側の終端へと移動します。
疑わしいページ (suspect page)	障害発生のため、リカバリが疑わしいと判断されたページ。疑わしいページは特別なプロシージャで強制的にオンラインにしないかぎり、ユーザはアクセスできません。
え	
エイリアス (alias)	データベースにおいて、特定の Adaptive Server ユーザを別のユーザとして認識させるための別名。

エクステント (extent)	テーブルやインデックスに領域が必要なとき、 Adaptive Server は常に、エクステントと呼ばれる 8 つのページのブロックをオブジェクトに割り付けます。エクステントのサイズは、サーバが使用している論理ページ・サイズ (2、4、8、16K) によって異なります。
エクステントの横取り (extent stealing)	パーティションがデバイス上の領域を使いいきり、エクステントが別の物理デバイスに割り付けられること。エクステントの横取りは、 <code>insert</code> コマンドまたはバルク・コピーを使ってテーブルにデータを挿入している場合や、クラスタード・インデックスを作成しているときに発生します。
エクステント割り付け (extent allocation)	テーブルまたはインデックスに新しい領域を割り付ける方法。多くの場合、新しい領域は一度に 1 ページずつ割り付けられます。ただし、大きな領域を割り付ける必要のあるコマンドでは、エクステント割り付けを実行し、一度に 8 ページ割り付けることができます。
枝分かれの多いツリー・プラン (bushy tree plan)	いくつかのジョイン演算を含み、そのジョイン演算がさらにジョイン演算である 2 つ以上の直接の子を持っている クエリ・プラン (query plan) 。通常、枝分かれの多くないプランは「左側の深いツリー」と呼ばれます。「左側の深いツリー」の右側の子は <code>scan</code> 演算子 (ジョイン・リスト) で、ジョイン・リスト内のテーブルはリニア順でジョインされます。
枝分かれの多い並列処理 (bushy parallelism)	複数の CPU が複雑な クエリ・プラン (query plan) の異なるサブプランを並列に実行するときに発生します。
エラー処理 (error handling)	Transact-SQL プログラマが利用できる処理。コードに基づいてエラーやエラー・メッセージを表示します。
エラー・ステータス番号 (error state number)	Adaptive Server エラー・メッセージに付属する番号。エラーの発生した Adaptive Server コードの行をユニークに区別するために使用します。
エラー・メッセージ (error message)	Adaptive Server が生成するメッセージ。通常、 Adaptive Server がエラー条件を検出すると、ユーザの端末に送られます。
エラー・ログ (error log)	重大な エラー・メッセージ (error message) 、およびデータベースの起動とリカバリの結果を保管するファイル。
演算子 (operators)	2 つの値に対して働き、第 3 の値を生成する記号。「 比較演算子 (comparison operator) 」、「 論理演算子 (logical operator) 」、「 算術演算子 (arithmetic operator) 」参照。
エンジン (engine)	共有メモリ内の他の Adaptive Server エンジンと通信可能な Adaptive Server 実行ファイルのインスタンス。ユニプロセッサ・マシンで稼動する Adaptive Server のエンジンは、常にエンジン 0 の 1 つだけです。マルチプロセッサ・マシンで稼動する Adaptive Server では、1 つ以上のエンジンを使用できます。 サーバ・エンジン (server engine) ともいいます。
エンジン・グループ (engine group)	1 つまたは複数のエンジン群。

エンタープライズ・データ (enterprise data)	ネットワーク・システム上にあるデータ。企業データは Sybase サーバに保存するか、異環境のデータとして保存できます。
エンティティ (entity)	ユニークな ID で識別でき、データベース・ページによって実現されるデータベースまたはデータベース・オブジェクト。エンティティの例としては、データベース pubs2、データベース pubs2 用のログ、データベース pubs2 内のテーブル titles 用のクラスタード・インデックス、データベース pubs2 内のテーブル authors があります。データベースの論理設計 (logical design) では、まずエンティティの識別を行います。
お	
応答時間 (response time)	Adaptive Server に対する Transact-SQL クエリの送信など、単一のタスクが終了するまでの所要時間。「初期応答時間 (initial response time)」参照。
オーバフロー・ページ (overflow page)	ユニークでないクラスタード・インデックスのあるテーブルのデータ・ページ。このページには重複したキーを持つローだけが含まれています。キーの値はチェーン上の前ページの最後のキーと同じです。オーバフロー・ページを直接指すインデックス・ページはありません。
置き換え更新 (in-place update または update in place)	直接更新 (direct update) オペレーションの 1 つ。置き換え更新では、データ・ローはデータ・ページに移動しません。「ページ上更新 (on-page update)」と「削除/挿入型直接更新 (delete/insert direct update)」参照。
置き換え更新 (Update in place)	「置き換え更新 (in-place update または update in place)」参照。
オブジェクト (object)	情報が含まれた、または情報を受け取る受動的エンティティ。その中の情報は変更できません。Adaptive Server のオブジェクトには、ロー、テーブル、データベース、ストアド・プロシージャ、トリガ、デフォルト、ビューがあります。「データベース・オブジェクト (database object)」参照。
オブジェクト・アクセス・パーミッション (object access permission)	特定のデータベース・オブジェクトにアクセスするための特定のコマンドの使用の制限。オブジェクト・アクセス・パーミッションは、オブジェクトの所有者によって付与されたり取り消されたりします。
オブジェクト・アロケーション・マップ (object allocation map: OAM)	テーブルやテーブル内のインデックス上の 1 ページ以上のページ。領域を使用するオブジェクトの各アロケーション・ユニット (allocation unit) 用のアロケーション・ページを指すポインタが含まれます。
オブジェクト・アロケーション・マップ・スキャン (object allocation map (OAM) scan)	最初にテーブルの OAM ページ、次にアロケーション・ページ、最後にデータが保存されているデータ・ページを読み込んで、データにアクセスする方法。

オブジェクト・アロケーション・マップ・ページ (object allocation map (OAM) pages)	各テーブルとテーブルのインデックスには、OAM (オブジェクト・アロケーション・マップ) ページがあります。OAM ページにはテーブルやインデックスに割り当てられたページに関する情報が格納され、インデックスやテーブルに新しいページが必要になった場合にチェックされます。
オブジェクト作成のパーミッション (object creation permission)	オブジェクトを作成するコマンドの使用の制限。これらのパーミッションを付与できるのは、システム管理者とデータベース所有者だけです。
オブジェクト所有者 (object owner)	オブジェクトを作成したか、または所有権を与えられたかのどちらかによってオブジェクトを所有するユーザ。たとえば、システム管理者はユーザをデータベース所有者に指定でき、データベース所有者はユーザをテーブル所有者に指定したり、テーブルを作成するパーミッションをユーザに付与することができます。
オブジェクトのパーミッション (object permission)	特定のコマンド (データ修正 (data modification) コマンド、select、truncate table、execute コマンド) の使用を、特定のテーブル、ビュー、カラムに制限するパーミッション (permission)。「コマンド・パーミッション (command permission)」参照。
オフセット・テーブル (offset table)	データの終わり、および、ページにおけるローのバイト・ロケーションを示すポインタを保存するインデックス・ページのフィールドのセット。全ページロック・テーブルのインデックスはオフセット・テーブルを持ちません。それ以外のすべてのインデックスとすべてのデータ・ページはオフセット・テーブルを持ちます。
オプティマイザ (optimizer)	クエリとデータベース・オブジェクトを解析し、適切なクエリ・プラン (query plan) を選択する Adaptive Server コード。Adaptive Server オプティマイザは、コストベースのオプティマイザです。テーブル・アクセスのそれぞれの変化のコストを、CPU コストと I/O コストの面から見積もります。
オペレータ (Operator)	データベースのバックアップおよびリストアをサーバワイドで行うことのできる役割。「役割 (roles)」参照。
オペレーティング・システム (operating system)	ファイルの作成、プログラムの実行、文書の印刷などのタスクが実行できるようにユーザのコマンドをコンピュータに伝える一群のプログラム。
オペレーティング・システム・ファイル (operating system file)	オペレーティング・システム (operating system) によって命名され、認識されるデータ・コレクション。Adaptive Server データはオペレーティング・システム・ファイルには保存されませんが、バルク・コピー (bulk copy) 操作を使用するとオペレーティング・システム・ファイルにエクスポートできます。
オンライン・トランザクション処理 (online transaction processing: OLTP)	最小限のリソースを使用するクエリを含んでいる、短時間のトランザクションを多数処理することを特徴とするトランザクション処理。

か

カーソル (cursor)	指定したテーブルから 1 つ以上のローを検索し、ローを個々に変更や削除するときに使う名前付き select 文。カーソルは、 カーソル結果セット (cursor result set) と カーソル位置 (cursor position) の 2 つの部分で構成されています。
カーソル位置 (cursor position)	カーソルの現在のロー位置を表します。 delete や update など、カーソルをサポートする文を使ってその行を明示的に参照できます。現在のカーソル位置は、 fetch で変更できます。これによって、 カーソル結果セット (cursor result set) の方へ、現在のカーソル位置が 1 行または複数行下に移動します。
カーソル結果セット (cursor result set)	カーソルに対応する select 文を実行して得られるローのセット。
カーソル・スキャン (cursor scan)	カーソル結果セット (cursor result set) を生成するプロセス。
カーソル・スコープ (cursor scope)	カーソル (cursor) を使うコンテキスト。カーソルの存在は、そのスコープ、すなわち特定のユーザ・セッション内、ストアド・プロシージャ内、トリガ内によって異なります。
カーソルの安定性 (cursor stability)	Adaptive Server が現在のカーソル・ローのあるベース・テーブル・ページに共有 ロック (lock) を設定している ロック・レベル (locking level) または 独立性レベル (isolation level) 。(フェッチが行われた結果) カーソルがページの外に出ないかぎり、そのページはロックされたままになります。ベース・テーブルにインデックスがあると、対応するインデックス・ページにも共有ロックが設定されます。
カーネル (kernel)	Adaptive Server とオペレーティング・システム間のインタフェースとしての役割を果たす Adaptive Server 内のモジュール。カーネルは Adaptive Server のクライアントに関連するタスクを管理します。コンテキストの切り替えのためにマルチスレッド環境で必要なタスクのステータスを追跡します。
開発ツール (development tool)	PowerBuilder のように、 Adaptive Server データベースにアクセスするための、特別な GUI アプリケーションの構築をサポートするソフトウェア。
外部キー (foreign key)	論理的に別のテーブルの プライマリ・キー (primary key) ・カラムに從属するテーブル内のキー・カラム。また、値が他のテーブルの プライマリ・キー に一致する必要のあるカラム (または複数のカラムの組み合わせ) です。
外部クエリ (outer query)	サブクエリ (subquery) が含まれる文における主要 クエリ (query) の別名。
外部ジョイン (outer join)	一致ローと非一致ローの両方が返される ジョイン (join) 。演算子 *= と =* は、 ジョイン・カラム で一致があるかどうかに関係なく、1 つ目のテーブルと 2 つ目のテーブルにあるすべてのローが返ることを表します。
外部テーブル (outer table)	外部ジョイン (outer join) のテーブルの配置の説明。たとえば、左側のジョインでは、外部テーブルも左側のテーブルです。

外部ログイン (external login)	コンポーネント統合サービス・クライアントでリモート・サーバにログインするときを使う代替ログイン名とパスワード。sp_addexternlogin を使って作成します。
カウンタ (counter)	Windows NT パフォーマンス・モニタで知ることのできる測定可能なパフォーマンス項目。これは、パフォーマンスに関する統計的な情報を生成するメカニズムを表現するために使用される一般的な Windows NT 用語です。Adaptive Server は、Adaptive Server オブジェクトや Windows NT プラットフォーム上のイベントを測定およびレポートするために、一連の特殊なカウンタを管理します。
書き込みアクセス権 (write access)	ロー (row) の更新やローのテーブル (table) への追加などの、オブジェクトの書き込みパーミッション。
拡張ストアド・プロシージャ (extended stored procedure)	Adaptive Server から外部手続き型言語関数を呼び出すメカニズム。通常のストアド・プロシージャと同じ構文で外部ストアド・プロシージャを呼び出します。ただし、Transact-SQL 文を実行するかわりに ESP が手続き型言語コードを実行します。これはダイナミック・リンク・ライブラリ (DLL) や共有ライブラリでコンパイルされます。
拡張済み高可用性 フェールオーバー (extended high-availability (HA) failover)	Adaptive Server は接続時に HA 対応クライアントに対してフェールオーバー・アドレスのリストを提供します。これによって、複数のクライアントへのフェールオーバーでのディレクトリ・サービスまたは interfaces ファイル内の “HAFAILOVER” エントリの必要性がなくなります。
格納処理 (store operator)	完全にマテリアライズされたテーブルを作成する処理。通常は、再フォーマット方式のサポートに使用します。
格納領域管理プロパティ (space management property)	テーブル領域の割り当てまたは使用の管理に役立つテーブルまたはインデックスのプロパティの 1 つ。fillfactor、max_rows_per_page、exp_row_size、および reservepagegap があります。領域管理プロパティは、ユニーク・キーまたは主キー制約を生成して実行するコマンドにより指定されます。テーブルでは create table か alter table、インデックスでは create index か alter table の各コマンドが使用されます。領域管理プロパティの一部は sp_chgattribute によって変更が可能です。
仮数 (mantissa)	数値の小数部分。たとえば、1.654 の仮数は .654 です。
カスケード・オプション付き (with cascade option)	パーミッションを取り消すときのオプション。指定されたユーザやグループからパーミッションを取り消し、指定されたユーザやグループがパーミッションを付与していたユーザからもそのパーミッションを取り消します。
カスケード型削除 (cascading delete)	ほかのテーブルにある関連データに影響を与える削除操作。
カスケード・メニュー (cascading menu)	選択したプルダウン・メニュー項目の右側に表示されるサブメニュー。プルダウン・メニューの項目の隣に矢印が表示されていると、カスケード・メニューがあることを示しています。

仮想カラム (virtual column)	実行エンジンの演算子の出力に表示される、 テーブル (table) の永続的なカラムに直接マップされない カラム (column) 。通常は、 ジョイン (join) の一方の 式 (expression) 。
仮想計算カラム (virtual computed column: VCC)	指定された テーブル (table) や インデックス (index) ページに結果が格納されない 計算カラム (computed column) 。
仮想サーバ・アーキテクチャ (Virtual Server Architecture: VSA)	ハードウェア・リソースとオペレーティング・システム・リソースを効率的に利用することを目的としたデータベース・アーキテクチャ。Sybase における SMP 処理の実装です。VSA は Adaptive Server エンジンによって、完全な対称処理と Sybase 独自のデータベース・カーネルを実現します。これは、タスクのスケジューリングとディスクパッチなどの代表的なオペレーティング・システム機能を扱うマルチスレッド・オペレーティング・システムです。
カタログ・ストアード・プロシージャ (catalog stored procedure)	システム・プロシージャ (system procedure) の種類の 1 つ。テーブル形式の システム・テーブル (system table) からデータを返します。
カバード・クエリ (covered query)	「 インデックス・カバリング (index covering) 」参照。
カバリング (covering)	「 インデックス・カバリング (index covering) 」参照。
カラム (column)	エンティティ (entity) の 1 つの特性を記述するデータ値。カラムには、ローまたはレコード内の個々のデータ項目が含まれます。 フィールド (field) ともいいます。
カラムレベルの制約 (column-level constraint)	指定した カラム (column) の値の制限。 create table 文では、カラム名とデータ型のあと、区切りカンマの前に、カラムレベルの制約を指定します。
関係演算子 (relational operator)	2 つのオペランドを比較して、“5 < 7” (TRUE)、“ABC” = “ABCD” (FALSE)、“@value > NULL” (UNKNOWN) のような真理値を生成する演算子。
関係式 (relational expression)	以下のような形式のブール式または論理式。 <code>arith_expression relational_operator arith_expression</code> Transact-SQL は、関係式は TRUE、FALSE、または UNKNOWN を返します。式のどちらか、または両方で NULL と評価されると、結果は UNKNOWN になります。
監査 (auditing)	セキュリティ関連システムのアクティビティに関する記録操作。システムへの侵入やシステム・リソースの誤用を検出します。
監査証跡 (audit trail)	sybsecurity データベースに保管される監査レコード。
関数 (functions)	「 組み込み関数 (built-in functions) 」参照。

関数ベース・インデックス (function-based index)	1つ以上の式をインデックス・キーとして含む インデックス (index) 。
関連性 (relationship)	エンティティがどのように関連するかの説明。エンティティの関係を示す記述。データベースの 論理設計 (logical design) の基本段階は、識別したエンティティ同士の関係を明らかにすることです。「 エンティティ (entity) 」参照。
関連付けキー (association key)	関連付けキーがユーザ ID、グループ ID、およびクエリ・テキストの組み合わせで構成されるということは、特定のユーザに対して、1つの抽象プラン・グループにクエリ・テキストが同じでクエリ・プランが異なるクエリが2つ存在しないことを意味します。
き	
キー (key)	レコードを識別するために使用するフィールド。一般にテーブルの インデックス (index) フィールドとして使用します。
キー値 (key value)	インデックスが付けられた値。
キーワード (keyword)	Transact-SQL で排他的に利用するように予約されているワードやフレーズ。 予約語 (reserved word) とも呼ばれます。
キーワード句 (keyword phrase)	「 句 (clause) 」参照。
基本となるテーブル (underlying table)	「 ベース・テーブル (base tables) 」参照。
基本日付 (base date)	1900年1月1日。日付カラムに値を指定しない場合、 Adaptive Server が自動的に設定する日付です。
基本優先度 (base priority)	タスクまたはスレッド作成時の優先度。スレッドの実行優先度は常に基本優先度と同じかそれ以上であり、それ未満になることはありません。
キャッシュ (cache)	「 バッファ・キャッシュ (buffer cache) 」参照。
キャッシュ・ヒット率 (cache hit ratio)	Adaptive Server は多くのプロセスで、メモリ内キャッシュを使用します。キャッシュ・ヒット率は、必要なページまたは結果がキャッシュ内で発見された回数をパーセントで示します。データ・ページの場合、キャッシュ・ヒット率は、ディスク I/O を必要とする要求に対してデータ・キャッシュが提供したページ要求の割合をパーセントで示します。
キャッシュ方式 (cache strategy)	実行エンジンがクエリで参照される明示的なテーブルのページを処理するために使用するバッファ・キャッシュの特性や、クエリの内部処理に使用される一時的な情報についての、 オプティマイザ の指定。
強制 (forcing)	クエリやセッションで、 オプティマイザ によるクエリ・プランの選択を上書きするオプションを含みます。

共有ディスク・クラスタ (shared-disk cluster)	(クラスタの概念)すべてのインスタンスが、すべての共有ディスク上にあるすべてのデータに直接アクセスできるクラスタ構成。Cluster Edition では、すべてのインスタンスが、データベース・デバイスに直接アクセスでき、データベースの単一インストールを共同で管理します。
共有ロー・ロック (shared row lock)	読み込み処理のデータ・ローに対して取得されるロック・タイプ。他のトランザクションは、このローについて排他ロックを取得できませんが、共有ロックは取得できます。
共有ロック (shared lock)	読み込み操作でオブジェクトに設定されるロックの1つ。共有ロックは、他のトランザクションによるオブジェクトへの排他ロックを許可しませんが、共有ロックは許可します。共有ロックはテーブルやページに対して設定することができます。
<	
句 (clause)	Transact-SQL コマンドを特定の条件に合わせるためのキーワード (keyword) とオプションのセット。キーワード句ともいいます。
クエリ (query)	1つのSQL文またはSQL文のグループ。データベースのデータに対するアクセスや操作、またはその両方を行います。「データ検索 (data retrieval)」参照。
クエリ・オブジェクト (query object)	SQLクエリを作成し、その結果セットを管理するオブジェクト。
クエリ処理 (QP) 測定基準 (query processing (QP) metrics)	クエリ処理 (QP) 測定基準は、クエリの実行において経験的な測定基準値を識別して比較します。クエリが実行されると、この測定基準は、QP 測定基準での比較の基礎となる定義済みの測定基準のセットと関連付けられます。
クエリ・ツリー (query tree)	ユーザのクエリ (query) を表す内部ツリー構造。クエリ処理とコンパイルの大部分は、この内部データ構造の形態と構造に合わせて構築されます。ストアド・プロシージャ、ビュー、トリガ、ルール、デフォルトの場合、これらのツリー構造は、ディスクの sysprocedures テーブルに保存され、プロシージャやビューの実行時にディスクから読み込まれます。「ソース・テキスト (source text)」参照。
クエリ・プラン (query plan)	クエリ (query) の実行に必要な、順序が決められた手順。テーブルごとに選択したアクセス・メソッドとともに使用されます。
クエリ・プロセッサ (query processor)	クエリの実行コストを最小限に抑えるように設計されたプロセッサ。解析、正規化されたクエリを調べ、データベース・オブジェクトに関する統計を使用して、最適なクエリ・プランを決定します。
クォーラム・デバイス (quorum device)	(クラスタの概念)このデバイスは、クラスタの名前、クラスタ内のインスタンスの名前、ノードの数、およびその名前など、クラスタを定義するための重要な情報を提供します。また、クォーラム・デバイスには、クラスタ内のインスタンスに関するステータス情報が保存され、クラスタのメンバーシップを定義します。

区切り識別子 (delimited identifiers)	二重引用符で囲んだオブジェクト名。これによってオブジェクト名に対する制限を回避できます。
組み込み関数 (built-in functions)	1つまたは複数の引数 (argument) を使って結果を返す、さまざまな種類の関数。組み込み関数には、集合関数 (aggregate function)、算術関数 (mathematical function)、システム関数 (system function)、文字列関数 (string function)、テキストおよびイメージ関数 (text and image function)、日付関数 (date function)、データ型変換関数 (datatype conversion function) があります。
クライアント (client)	クライアント/サーバにおけるユーザ側の配置。サーバに対して、またはクライアント・ソフトウェアを実行しているマシンに対して呼び出しを行うソフトウェアを指すこともあります。
クライアント・カーソル (client cursor)	Open Client 呼び出しまたは Embedded SQL を介して宣言されるカーソル。Open Client は、Adaptive Server から返されるローを追跡し、アプリケーション用にそれらのローをバッファします。クライアント・カーソルの結果セットの更新や削除ができるのは、Open Client 呼び出しだけです。
クライアント・タスク (client task)	クライアント要求に対応するために生成されるスレッド。
クライアント・マイグレーション (client migration)	(クラスタの概念) 確立済みのクライアント接続のインスタンス間での移動。クライアントは、移動元のインスタンスから移動先のインスタンスに移動します。この移動はアプリケーションに検知されません。クライアント・マイグレーションは、負荷の動的な分散と論理クラスタ・フェールバックなどの管理アクションに使用されます。
クライアント/サーバ・アーキテクチャ (client/server architecture)	コンピュータ・システム・アーキテクチャ。このアーキテクチャでは、クライアントによるサービスの要求およびサーバによるサービスの提供が行われます。これによって、各マシンを最適な専門タスクに特化させることができます。
位取り (scale)	numeric データ型や decimal データ型で小数点の右側に値を保存できる最大桁数。位取りは、精度 (precision) と同じまたはそれより少ない桁数でなければなりません。
クラス (class)	Java では、クラスによってオブジェクトが定義されます。オブジェクトはクラスをインスタンス化したものです。ユーザ・インタフェースを備えたアプリケーションにはフォームがあり、フォームとは設計環境に対する対応を備えたクラスです。
クラスタ (cluster)	1つのシステムとして動作するネットワーク上の同機種間ノードのコレクション。ノードには、それぞれ独自の CPU とメモリがあります。すべてのノードは、プライベートで高速な通信経路を使用して他のノードと通信します。
クラスタード・インデックス (clustered index)	物理順序と論理 (インデックス) 順序が同じであるインデックス。クラスタード・インデックスのリーフ・レベルはデータ・ページ自身を表します。1つのテーブルには1つのクラスタード・インデックスしか設定できません。

クラスタウェア (clusterware)	(クラスタの概念) 共有ディスク・クラスタを有効にする Adaptive Server に搭載されている Sybase ソフトウェア。
クラスタ・ロック・マネージャ (CLM : cluster lock manager)	(クラスタの概念) クラスタ用の分散ロック・サービスを提供するサーバモジュール。CLM を使用すると、バッファ、グローバル・オブジェクト、およびメタデータをインスタンス間で共有できます。
繰り返し可能読み出し (repeatable read)	ほかのトランザクションで行われた変更によって結果に影響の出ないクエリ。読み込んだページまたはローはトランザクションが終わるまでロックされるので、他のトランザクションはそのページやローを更新できません。また、トランザクション独立性レベル 2 ともいいます。このレベルでは、トランザクション独立性レベル 3 で提供される幻保護は提供されません。
繰り返しサブクエリ (repeating subquery)	「 関連サブクエリ (correlated subquery) 」 参照。
繰り返し不可能読み出し (nonrepeatable reads)	このような読み出しは、あるトランザクションがローを読み込み、2 番目のトランザクションがこのローを修正する場合に発生します。2 番目のトランザクションがその変更をコミットした場合、最初のトランザクションによる後続の読み込みは元の読み込みと異なる結果になります。
グループ (group)	データベース内のオブジェクトや操作に対するパーミッションが割り当てられた、ユニークな名前の付いたユーザの集まり。
グループ化された集合 (grouped aggregate)	「 ベクトル集合 (vector aggregate) 」 参照。
グループ化されていない集合 (ungrouped aggregate)	「 スカラー集合 (scalar aggregate) 」 参照。
グローバル・インデックス (global index)	分割されたテーブルのインデックス。グローバル・インデックスは、インデックスとテーブルの分割方式が異なる場合に作成され、グローバル・インデックスのインデックス・リーフ・ページが複数のパーティションを指すようにします。
グローバル統計 (global statistics)	テーブル (table) のすべてのデータ値に適用される統計。
グローバル変数 (global variable)	Adaptive Server が進行ベースに基づいて更新するシステム定義の 変数 (variable) 。たとえば、 @@error にはシステムが生成した最新のエラー番号が保管されます。「 ローカル変数 (local variable) 」 参照。
け	
計算カラム (computed column)	同じローの通常のカラム、関数、算術演算子、パス名などの式によって定義されるカラム。
計算カラム・インデックス (computed column index)	1 つ以上の計算カラムをインデックス・キーとして含むインデックス。

ゲートウェイ (gateway)	クライアントとデータ・ソース間で、言語の翻訳、 データ型 (datatype) の変換、プロトコル変換を行う中間ソフトウェア。
権限 (authorization)	「 役割 (roles) 」参照。
権限 (privilege)	「 パーミッション (permission) 」参照。
言語カーソル (language cursor)	Open Client を使用せずに SQL 内で宣言するカーソル。 Adaptive Server カーソルの場合、Open Client はカーソルをまったく認識せず、結果は通常の <code>select</code> と同じフォーマットでクライアントに返信されます。
言語グループ (language group)	1つの文字セットでカバーされる言語。言語グループは複数言語、または1つの言語で構成されます。
言語モジュール (Language Module)	ローカライゼーション・ファイルなどのファイル・セット。代替言語、ソート順序、 Adaptive Server 用の文字セットを提供します。
現在の値 (current value)	現在のローを更新する直前の変数値。
検索エンジン (search engine)	クエリ・オプティマイザのコンポーネントの1つ。代替実行プランを生成して評価し、最適なプランを選択します。検索エンジンには、 検索基準 (search criteria) 、 検索領域 (search space) 、 検索方式 という3つのキー・コンポーネントがあります。
検索基準 (search criteria)	プランの生成に使用する最適化テクニックを決定するために使用される、ユーザ指定またはシステム定義の基準。
検索更新 (searched update)	カーソルがアクティブな間に、カーソルを使わずに実行する更新。つまり、 <code>where current of</code> 句は使用しません。「 位置付け更新 (positioned update) 」参照。
検索削除 (searched delete)	カーソルがアクティブな間に、カーソルを使わずに実行する削除。つまり、 <code>where current of</code> 句は使用しません。「 位置付け削除 (positioned delete) 」参照。
検索方式 (search strategy)	検索エンジン (search engine) のモジュールの1つ。特定の 検索領域 (search space) を生成し、その検索領域内に含まれる代替プランの中からプランを選択します。
検索領域 (search space)	検索エンジンがプランを選択するときに検討対象となる、プランの包括的なセット。
検査制約 (check constraint)	ユーザがテーブルのカラムに挿入できる値を限定する <code>check</code> コマンドに指定する制約条件。 <code>check</code> 制約では、 <code>search_condition</code> を指定し、値はテーブルに挿入される前に渡される必要があります。
限定述語サブクエリ (quantified predicate subquery)	<code>in</code> 、または <code>any</code> や <code>all</code> で変更された比較演算子で導入されたりリストで使用するサブクエリ。限定述語サブクエリは、0以上の値を返します。このタイプは、 <code>exists</code> とともに指定される存在テストとしても使用されます。

こ

高可用性 (high availability)	システムのダウン時間を減らすように設計されたシステム。
交換処理 (exchange operator)	データ・ストリームに適用して、ストリームの並列度やデータ・セマンティックを変更できる処理。再分割に利用されます。交換処理には、プロデューサ・サイドとコンシューマ・サイドの両方があります。プロデューサ・タスクでは、交換処理の下で関係演算子の複製が実行されます。コンシューマ・サイドでは、コンシューマ処理に必要なのと同じ数の複製で実行されます。
更新 (update)	insert、delete、truncate table、update 文を使ったデータの追加、削除、変更。
更新ロー・ロック (update row lock)	更新操作や削除操作のスキャン段階でデータ・ローに対して行われる ロック (lock) の 1 つ。このロックが設定されたローに対して、他のトランザクションからの更新ロックや 排他ロック (exclusive lock) は実行できませんが、 共有ロック (shared lock) は可能です。
更新ロック (update lock)	更新操作や削除操作のスキャン段階でオブジェクトに対して行われる ロック (lock) の 1 つ。このロックが設定されたオブジェクトに対して、他のトランザクションからの更新ロックや 排他ロック (exclusive lock) は実行できませんが、 共有ロック (shared lock) は可能です。更新ロックはページのみに設定できます。
候補キー (candidate key)	プライマリ・キー (primary key) または 一意性制約 (unique constraint) カラム。1 つのテーブルに複数の候補キーを設定できます。
構造化問合せ言語 (Structured Query Language: SQL)	リレーショナル・データベースとの通信に使う言語。いくつかの標準化機関によって設定された標準サブジェクトです。
コーディネーティング・プロセス (coordinating process)	並列ソートでは、コーディネーティング・プロセスはコンシューマ・プロセスの結果を 1 つの結果セットに結合します。クエリの場合、結果セットが最終のソート済みデータになります。create index 文では、コーディネーティング・プロセスによってサブインデックスが 1 つの最終インデックスに結合されます。
コード生成 (code generation)	オプティマイザの最適なプランの表示を実行エンジンのプランの表示に変換する、クエリのコンパイルのメカニズム。オプティマイザが使用するプランの表示は、競合する最適なプランの効率的な比較のために設計されていますが、実行プランが使用するプランの表示は、効率的な実行のために設計されています。
コード・セット (code set)	「 文字セット (character set) 」参照。
互換データ型 (Compatible datatypes)	暗黙的比較または明示的比較のために自動的に変換されるデータ型。
国際化 (internationalization)	アプリケーションの複数言語、文化的な規則への対応を可能にするプロセス。

**国際文字セット
(national character
sets)**

「ネイティブの文字セット (native character set)」参照。

**コストベースの排除
(cost based pruning)**

コストの高いほかのサブプラン分析手法の代わりにサブプランの見積もりコストを使用する、クエリの最適化テクニック。

**固定ロー ID (fixed
row ID)**

ローに関連付けられているロー ID (row ID: RID) が変わらないようにする方法。ローのオフセット・テーブル (offset table) は、ページ内でのローの移動によって変わることがありますが、ロー番号自体は変わりません。ローの拡張によってローが移動する必要がある場合は、ローは新規ページにマイグレートし、転送先アドレス (forwarding address) はローの元の場所に格納されます。クラスタード・インデックスが作成されたり、reorg rebuild コマンドが実行される場合を除いて、ロー ID は永久に変わりません。ローがページから物理的に削除された後は、ロー ID は再使用されます。

**子ページ・ポインタ
(child page pointer)**

キー値を含むインデックスの最低レベルの次のレベルを指すインデックス内のポインタ。子ページ・ポインタはキーとロー ID で構成されます。

コマンド (command)

コンピュータによって実行される操作を指定する命令。それぞれのコマンドや SQL 文は、insert など、実行する基本の演算を示すキーワードで始まります。多くの SQL コマンドには、1 つ以上のキーワード句 (keyword phrase) または句 (clause) があります。これらは特定の要求に合うようにコマンドを管理します。

**コマンド・ターミネータ
(command terminator)**

Adaptive Server にバッチ処理を送信するバッチの終わりを示すシグナル。

**コマンド・ダイアログ・ボックス
(command dialog
box)**

実行方法に関する情報を必要とするコマンドの実行時に開かれるダイアログ・ボックス。

**コマンド・パーミッション
(command permission)**

コマンドに適用するパーミッション (permission) のこと。「オブジェクトのパーミッション (object permission)」参照。

**混合負荷 (mixed
workload)**

リレーショナル・クエリには、大きく分けて、OLTP 環境で実行される単純なトランザクション・クエリと、DSS 環境で実行される複雑なクエリとがあります。運用環境では、トランザクション・クエリか複雑なクエリのいずれかを、同時または時間をずらして実行するように、データベース・システムを設定します。これらの両方のクエリをサポートするシステムを、「混合負荷」システムと呼びます。負荷の種類を常に予測できるとは限らないため、すべての種類の負荷を効率的にサポートできるように、OLTP クエリと DSS クエリの両方を同じ設定のデータ処理システムでサポートすることが重要です。

**コンシューマ・プロセス
(consumer
process)**

並列ソートにおいて、コンシューマ・プロセスは、プロシージャ・プロセスから受け取ったデータの個別の範囲を同時にソートします。コンシューマ・プロセスは、ソートしたデータをコーディネーティング・プロセス (coordinating process) に渡します。

コンテキストで区別されるプロテクション (context-sensitive protection)	ユーザの ID にもとづいて特定のパーミッションや権限を与えるプロテクション機能。このタイプのプロテクション機能は、ビューや <code>user_id</code> 組み込み関数で実行されます。
コンテナ (container)	イベント・プロセッサが、イベント本文を構成するバイトのシーケンスを解釈する方法を決定します。SEEB は、バイナリ・イベント、SQL ローセット・イベント、SQL 入力プロパティ・イベント、テキスト・イベント、 <code>rvmsg</code> 、集合、Java オブジェクト、プロパティのコンテナに対応しています。
コンパートメント (compartment)	<code>sensitivity</code> レベルを作成するために分類で使用する非階層値の 1 セット。コンパートメントは、一般にトピックやワーク・グループを表します。
コンパイル (compilation)	クエリ・テキストを分析して、実行エンジンに提供するクエリ・プランを作成するクエリ処理の段階。
コンパイル済みオブジェクト (compiled object)	検査制約、宣言デフォルト、ルール、ストアド・プロシージャ、トリガ、ビューなど、 <code>sysprocedures</code> テーブル内にエントリが必要なオブジェクト。これらのオブジェクトは ソース・テキスト (source text) で記述します。 Adaptive Server は、各データベースに関する重要な情報を取得したり、ユーザによるデータのアクセスや操作をサポートするために、コンパイル済みオブジェクトを使用します。
コンパニオン・サーバ (companion server)	高可用性システムの各 Adaptive Server をコンパニオンと呼びます。
コンポーネント (component)	パッケージは複数のコンポーネントで構成されています。コンポーネントは、1 つまたは複数のメソッドで構成されるアプリケーション・オブジェクトです。
コンポーネント統合サービス (Component Integration Services : CIS)	Adaptive Server の機能を拡張し、相互運用性を提供します。OmniConnect の中心的な相互運用性機能です。コンポーネント統合サービスでは、 Adaptive Server と OmniConnect で統一された形式の企業データをクライアント・アプリケーションで表示させ、企業規模のデータ・ソースのロケーションを透過的にします。
さ	
サーバ・エンジン (server engine)	「 エンジン (engine) 」参照。
サーバ・カーソル (server cursor)	ストアド・プロシージャ (stored procedure) 内で宣言される カーソル (cursor) 。ストアド・プロシージャを実行するクライアントには、これらのカーソルの存在は認識されません。 <code>fetch</code> に対してクライアントに返される結果は、通常の <code>select</code> から返される結果とまったく同じです。
サーバ・クラス (server class)	コンポーネント統合サービスで使うリモート・アクセスを定義するインタフェース仕様。サーバ・クラスごとに、そのサーバと同じクラスのリモート・サーバの対話を処理するための別々のアクセス・メソッドがあります。各サーバ・クラスには、システム管理者とプログラマがサーバをリモート・データ・アクセス用に設定するための一意の特性が用意されています。それらの特性を次に示します。

- ローカル・テーブル (プロキシ・テーブル) と外部オブジェクトとのマッピング
- クラスやアクセス・メソッド固有のデータ型変換
- クラス別注意事項
- 必要に応じて、クラスに適用される Transact-SQL 文の制約

サーバ証明書 (server certificate)

サーバは、サーバ証明書を所有することで、サーバに接続しようとするクライアントに対して、自分がサーバであることを示します。

サーバ・ユーザ ID (server user ID)

[Adaptive Server](#) がユーザを識別する ID 番号。

最適化タイムアウト (optimization timeout)

コンパイル時間が指定した基準を超えたときに、オプティマイザが現在の最適プランよりも優れたプランの検索を終了するメカニズム。その場合、現在の最適プランが実行エンジンでのクエリの処理に使用されます。

最適化目標 (optimization goal)

最適化テクニックを選択するとき考慮される、一連のユーザ定義の目標。この目標を指定することにより、特定のクエリまたはアプリケーションに適したプランを生成できます。

最適化ルール (optimization rules)

オプティマイザが最良の最適化プランを決定するときは、見積もりコストに基づいて決定する場合がほとんどですが、クエリの述部やテーブルの特性に基づいて決定する場合があります。たとえば、スター・ジョイン以外の2つのテーブル間では、ジョイン述部を使用してオプティマイザがほかのジョイン順 (順序) を評価しないようにする方法が最適です。

サイト・ハンドラ (site handler)

ローカル・サーバからリモート・サーバへの RPC 送信のデフォルトの方法。サイト・ハンドラは、RPC が必要とするローカルおよびリモート・サーバ間で、単一の物理的な接続および複数の論理接続を確立します。

再フォーマット方式 (reformatting strategy)

有効なインデックスのない大きなテーブル上のジョイン・クエリを [Adaptive Server](#) で処理するための方式。Adaptive Server は内部テーブルのジョイン・カラムにテンポラリ・クラスタード・インデックスを作成し、このインデックスを使ってローを検索します。Adaptive Server はこの方式のコストと代替方式 (テーブル・スキャン) のコストを見積もり、コストのかからない方を選択します。

先書きログ (write-ahead log)

トランザクション・ログなど、ユーザがデータベースを変更する文を発行すると、[Adaptive Server](#) が自動的に書き込むログ。文による変更がすべてログに記録された後で、その変更がキャッシュ内のデータページのコピーに書き込まれます。

作業領域 (workspace)

1つまたは複数の [ターゲット \(target\)](#) の集まり。作業領域には、アプリケーション全体のすべてのターゲットを含めることができます。

削除/挿入型直接更新 (delete/insert direct update)

[直接更新 \(direct update\)](#) オペレーションの1つ。ローは元のロケーションから削除され、新しいロケーションに挿入されます。

サスペンド・コンパニオン・モード (suspended companion mode)	コンパニオン・モードがサスペンドされた後の Adaptive Server のモード。このモードの間、Adaptive Server はフェールオーバーできず、もう片方の Adaptive Server とは独立して稼働します。
サブクエリ (subquery)	他の select、insert、update、delete 文や別のサブクエリの内部にネストされる select 文 (statement)。
サブジェクト (subject)	ユーザのかわりに動作するサーバ・プロセス。 Adaptive Server の実行中には、サブジェクトには、ユーザ、ストアド・プロシージャ (stored procedure)、trusted トリガ (trigger) が含まれます。
サロゲート・ペア (surrogate pairs)	2つのコード値のシーケンスから成る1つの抽象文字の、コード化された文字表現。サロゲート・ペアは、 2^{20} バイトのコード値を Unicode 標準に追加できるように設計されています。UTF-16 エンコーディングだけに適用されます。
算術演算子 (arithmetic operator)	SQL 文で算術式 (arithmetic expression) を作成できるようにするための記号。加算 (+)、減算 (-)、除算 (/)、乗算 (*) は、数値カラムに使用できます。モジュロ (%) は、integer データ型でのみ使用可能です。「 比較演算子 (comparison operator) 」参照。
算術関数 (mathematical function)	算術データの演算に必要な共通の値を返す関数。
算術式 (arithmetic expression)	数値オペランドだけの式。数値を1つだけ返します。Transact-SQL では、オペランドには任意の Adaptive Server 数値データ型、すなわち関数、変数、パラメータ、そのほかの算術式が使えます。 数値式 (numeric expression) ともいいます。
参照整合性 (referential integrity)	データの整合性を管理するルール。特にさまざまなテーブル上の プライマリ・キー (primary key) と 外部キー (foreign key) の関係を管理します。 Adaptive Server は、ユーザ定義のトリガ (trigger) や 参照整合性制約 (referential integrity constraint) に関する参照整合性を処理します。
参照整合性制約 (referential integrity constraint)	「参照元」テーブル (制約を定義するテーブル) に挿入されるデータは、「参照先」テーブルに一致する値がなければならないとする条件。参照元テーブルに一致する値がある参照先テーブルに対して、ローの削除やカラム値の更新はできません。また、参照元テーブルが削除されるか、または参照整合性制約が削除されるまで、参照先テーブルは削除できません。
し	
シータ・ジョイン (theta join)	ジョインの条件として 比較演算子 (comparison operator) を使う ジョイン (join) 。
式 (expression)	値を返す計算、カラム・データ、組み込み関数、またはサブクエリ。
式サブクエリ (expression subquery)	修飾されていない比較演算子で始まるサブクエリのタイプ。1つの値のみを返し、SQL で式が使用できる場所であれば、どこでも使用できます。

識別子 (identifier)	テーブル名やカラム名などのデータベース・オブジェクト (database object) を区別するための文字列。
施行時間 (time of enforcement)	Adaptive Server がリソース制限を適用している間のクエリ処理フェーズ。
システム関数 (system function)	データベース、特にシステム・テーブル (system table) から特別な情報を返す関数。
システム管理 (system administration)	Adaptive Server の物理記憶の管理、データベースの作成とバックアップ、ユーザ・アカウントの作成、パーミッションの付与、診断機能や修復機能の実行などのタスク。「データベース管理 (database administration)」参照。
システム管理者 (System Administrator)	ディスク記憶の管理、システム管理者役割の付与と取り消し、新しいデータベースの作成といった Adaptive Server システム管理の担当者。“sa” アカウントという単一ログイン (login) が、Adaptive Server のインストール時に作成されます。このログインは、システム管理者役割とシステム・セキュリティ担当者 (System Security Officer) 役割の両方で設定されます。個々の責任範囲を増やすには、“sa” アカウントを使用して役割 (roles) を個別のログインに割り当てます。
システム・セキュリティ担当者 (System Security Officer)	Adaptive Server で、監査、ログイン・アカウントのロックやロック解除、ユーザ定義役割の作成、パスワード管理などのセキュリティ関係の操作を管理するユーザ。「システム管理者 (System Administrator)」参照。
システム・データベース (system database)	Sybase が提供するデータベース。master、tempdb、model、sybssystemprocs などがあります。
システム・テーブル (system table)	データ辞書テーブルの一種。システム・テーブルは、Adaptive Server 全体と各ユーザ・データベースに関する情報を追跡します。master データベースには、ユーザ・データベースにはないシステム・テーブルがいくつか存在します。
システム・プロシージャ (system procedure)	sybssystemprocs に常駐するストアド・プロシージャ (stored procedure)。Sybase が提供します。システム・プロシージャは、システム・テーブル (system table) の情報を簡単に検索する方法や、データベース管理とシステム・テーブルの更新などのタスクを行うメカニズムを提供します。
システム・プロシージャ・テーブル (system procedure tables)	master データベース内のテーブル。内部システム値 (ステータス・ビットなど) を人間が判読できるフォーマットに変換するためにシステム・プロシージャ (system procedure) が使用します。
システム論理クラスタ (system logical cluster)	(クラスタの概念) 物理クラスタの論理的表現。システム論理クラスタは、物理クラスタの作成時に自動的に作成され、その物理クラスタと同じ名前が付けられます。論理クラスタ 'mycluster' は、システム論理クラスタです。
実行値 (run value)	現在使用中の設定パラメータの値。

実行カーソル (execute cursor)	単一の select 文のあるストアド・プロシージャによって定義された結果セットを持つクライアント・カーソルのサブセットとなるカーソル。ストアド・プロシージャではパラメータが使えます。パラメータの値は Open Client 呼び出しを通じて送信します。
実行クラス (execution class)	各エンティティのパフォーマンスの稼働条件に基づいた、アプリケーション、ログイン、ストアド・プロシージャの分類。実行クラスはそのクラスに属するオブジェクト関連する一連の実行属性によって定義されます。
実行時調整 (runtime adjustment)	ランタイムのクエリ・プランで指定されたワーカー・プロセス数が使用できない場合に、より少ない数のワーカー・プロセスでクエリを実行できるように、 Adaptive Server が作成する調整されたクエリ・プラン。
実行スレッド (thread of execution)	「ワーカー・プロセス (worker process)」参照。
自動リカバリ (automatic recovery)	Adaptive Server を再起動するたびに実行される処理。これによって Adaptive Server が停止する前に終了したトランザクションは次の段階に進み、終了していないトランザクションはロールバックされます。
射影 (projection)	演算子の出力で利用できる属性のセット。属性の最小限のセットで、このセットに含まれる各属性は、各演算子の親に必要です。
集合関数 (aggregate function)	クエリ結果の新しいカラムとして表示される合計値を作成する関数。Transact-SQL で利用できる集合関数には、平均 (avg)、最大 (max)、最小 (min)、合計 (sum)、項目数のカウント (count) があります。
修飾 (qualified)	データベース・オブジェクトの名前の前に、データベース名やオブジェクト所有者を指定して修飾すること。
従属 (dependent)	あるテーブル上のマスタ・データが別のテーブルのディテール・データと常に同期をとらないとデータベースの論理的整合性を保つことができない状態を、データは論理的にほかのデータに従属しているといいます。
重大度レベル番号 (severity level number)	エラー状態の深刻さを示す、10～24 の範囲の番号。
主キー制約 (primary key constraint)	プライマリ・キー (primary key) 制約は一意性制約であり、コンポーネント・キー・カラムでは null 値が使えません。テーブルごとに主キー制約は1つしか使えません。主キー制約は指定したカラムにユニーク・インデックスを作成し、このデータの整合性を保ちます。
述部 (predicate)	比較演算子 (comparison operator) をもとにした句。and、or、not には適用しません。
順序 (ordering)	インデックス・スキャンやソートによって返される結果セットに含まれる属性の特定の順序 (昇順または降順)。

ジョイン (join)	指定したカラム (column) 内の値を比較して複数のテーブル (table) 内のロー (row) をリンクする、リレーショナル・システムにおける基本的な操作。
ジョイン選択性 (join selectivity)	他のテーブルのローとジョインされる特定のテーブル (table) のロー (row) の推定数。ジョインするカラム (column) にヒストグラム (histogram) が存在する場合、ジョイン選択性はそのジョイン・ヒストグラムに基づいて決定されます。ヒストグラムがない場合は $1/N$ となります。ここで N は、ジョインする小さい方のテーブルのロー数です。
ジョイン・テーブル (joined table)	左右のテーブル間のジョインの結果。
ジョイン・ヒストグラム (join histogram)	最適化の実行中にだけ作成され、その後廃棄される中間ヒストグラム。等価ジョインされた2つのカラムのヒストグラムを取得し、ジョイン発生後のデータの分散をモデル化するヒストグラムを生成することによって作成されます。
ジョイン密度 (join density)	「総密度 (total density)」参照。
照合順 (collating sequence)	「ソート順 (sort order)」参照。
詳細 (detail)	別のテーブルのデータに論理的に従属しているデータ。たとえば、pubs2 データベースの salesdetail テーブルはディテール・テーブルです。sales テーブル内の各注文は、salesdetail 内の多くのエントリーに対応しています。salesdetail 内の各項目は、sales テーブル内のエントリーに対応していません。
証明書 (certificates)	あるエンティティ (サーバ、ルータ、Web サイトなど) の別のエンティティに対する一意性の確認に使用される公式ドキュメント。
初期応答時間 (initial response time)	クエリの最初の結果ローをユーザに返すまでの時間。クエリによっては、結果セットをすべて返すまでにかなりの時間がかかる場合でも、最初の応答時間が非常に短いことがあります。
初期化 (initialization)	「ディスクの初期化 (disk initialization)」参照。
ジョブ (job)	1 回の操作でデータベースに対して実行される一連のアクション (backups、update statistics、dump database など)。Job Scheduler と併用します。
所有権の連鎖 (ownership chain)	他のビューとテーブルの両方またはどちらかに対するビューの従属関係。および他のプロシージャ、ビュー、テーブルのすべてまたはいずれかに対するプロシージャの従属関係。
シングル・ユーザ・モード (single-user mode)	1 人のシステム管理者だけがログインでき、更新設定変数をオンにできる、Adaptive Server 起動モード。このモードは、master データベースをリストアする場合に使用してください。このオプションは <code>m_RUN_servername</code> ファイルを作成し、既存の <code>m_RUN_servername</code> ファイルを上書きします。

す

推移閉包 (transitive closure)	等価ジョイン (equijoin) で接続された属性のセット。
垂直並列処理 (vertical parallelism)	プランの 1 つのフラグメント内の複数の処理に対して、複数の CPU を同時に使用する機能。パイプライン並列処理 (pipelined parallelism) ともいいます。
水平並列処理 (horizontal parallelism)	分割並列処理と独立並列処理 (independent parallelism) は、水平並列処理 (horizontal parallelism) に分類されます。異なる記憶単位に配置された異なるデータセットで複数インスタンスの処理を実行する機能も、並列並行処理または分割並列処理 (partitioned parallelism) と呼ばれます。
数値式 (numeric expression)	「算術式 (arithmetic expression)」参照。
スカラ (scaler)	値を 1 つ生成する SQL 式。
スカラ集合 (scalar aggregate)	group by 句のない select 文で値を 1 つ生成する集合関数 (aggregate function)。集合関数がテーブル内のすべてのローについて実行される場合と、where 句で指定したロー・サブセットについて実行される場合のどちらにも適用できます。「ベクトル集合 (vector aggregate)」参照。
スキーマ (schema)	特定のスキーマ名とスキーマ権限識別子 (schema authorization identifier) に関連するオブジェクトの集まり。オブジェクトには、テーブル、ビュー、ドメイン、制約、表明、権限などがあります。スキーマは、create schema 文で作成されます。
スキーマ権限識別子 (schema authorization identifier)	すべてのオブジェクトは、対応するスキーマのスキーマ権限識別子によって所有されるか、または生成されています。
スキャン記述子 (scan descriptor)	クエリを行うテーブルのスキャンを管理する内部データ構造。特に他のテーブルを参照するテーブルに使用します。
スクリプト (script)	文字により記述するシステム。たとえば、ラテン語、日本語、アラビア語などの言語の記述形式を特徴付ける要素すべての集合です。
スクロール可能カーソル (scrollable cursor)	1 つまたは複数のローを前方または後方にフェッチできるカーソル。
スケジュール (schedule)	ジョブを実行および再実行する方法や時刻に関する定義。Job Scheduler と併用します。
スケジュール・ジョブ (scheduled job)	スケジュールにバインドされているジョブ。スケジュール・ジョブのみが実行されます。Job Scheduler と併用します。
スコープ (scope)	カーソル (cursor) やグローバル変数 (global variable) など、ある機能が使われるコンテキスト。Adaptive Server のスコープには、個々のユーザ・セッション、ストアド・プロシージャ (stored procedure)、またはトリガ (trigger) があります。

スター・スキーマ・ジョイン (star schema joins)	複数のディメンション・テーブル (dimension table) が中央ファクト・テーブル (fact table) にジョインされているクエリ。ディメンション・テーブル間にジョイン句はありません。また、中央ファクト・テーブルは、それにジョインされているディメンション・テーブルよりも大規模です。
スタンバイ・アクセス・サーバ (standby access server)	Adaptive Server 運用サーバを正確に複製したサーバ。運用サーバからスタンバイ・サーバへトランザクション・ログ・ダンプをロードして、ログ・ダンプが作成されると、スタンバイ・サーバが作成されます。スタンバイ・サーバは、メイン・サーバが故障した場合の緊急プロキシ・サーバになります。また、読み込み専用にして、レポート作成や意思決定支援のために使用することも可能です。
ステートフル・コンポーネント (stateful component)	メソッドを連続して呼び出す間、アクティブであり続けるコンポーネント。
ステートフル・セッション Bean (stateful session Beans)	セッション Bean のインスタンスは一時的なものであり、クライアントとの関係は 1 対 1 です。クライアントに代わってタスクを実行し、データベースに情報を格納します。ステートフル・セッション Bean は、データの累積が必要な、複雑なタスクを管理します。
ステートレス・コンポーネント (stateless component)	メソッドを呼び出すたびに非アクティブ化され、インスタンスのプールをサポートしているコンポーネント。コンポーネントのステータスは、トランザクションとアクティブ化の境目を越えるときにリセットされます。
ステートレス・セッション Bean (stateless session Beans)	セッション Bean のインスタンスは一時的なものであり、クライアントとの関係は 1 対 1 です。クライアントに代わってタスクを実行し、データベースに情報を格納します。ステートレス・セッション Beans が管理するタスクは、メソッド呼び出しとメソッド呼び出しの間にデータを格納しません。
ステーブル・モード (stable mode)	Adaptive Server が長期間存続できるシステム状態。たとえば日常の運用状態がそうです。
ステップ (step)	ディストリビューション (distribution) またはヒストグラム (histogram) に含まれる連続した値のペア。各ステップは、ステップの上限値と下限値の間の範囲のカラム値を持つローのセットに対応しています。セル (cell) ともいいます。
ストアード・プロシージャ (stored procedure)	Transact-SQL 文 (statement) とオプションのフロー制御文の集まりに名前をつけて保存したもの。「システム・プロシージャ (system procedure)」参照。
ストライプ化 (striping)	「ダンプ・ストライピング (dump striping)」参照。
スノーflake・スキーマ・ジョイン (snowflake schema joins)	複数のディメンション・テーブルがローカルなファクト・テーブル (fact table) にジョインされ、それらのファクト・テーブルが 1 つの中央ファクト・テーブルにジョインされているクエリ。ディメンション・テーブル (dimension table) 間にジョイン句はありません。また、各ファクト・テーブルは、それにジョインされているディメンション・テーブルよりも大規模です。

スピンロック (spinlock)	現在ほかのプロセスが使用中のリソースをあるプロセスがアクセスするのを防ぐための、単純な内部ロック・メカニズム。そのリソースにアクセスしようとするすべてのプロセスは、ロックが解放されるまで待つ (つまり「スピンする」) 必要があります。スピンロックでは、データ・キャッシュなどの内部データ構造を保護します。
スループット (throughput)	指定の時間内に終了する作業量。通常は 1 秒当たりのトランザクション数 (TPS) で表します。
スレッシュホールド (threshold)	トランザクション・ログのバックアップに必要と思われるログ・ページ数の見積もりと、スペース量がその値を下回ったときに実行される動作。
スレッド (thread)	「ワーカー・プロセス (worker process)」参照。
せ	
正規化規則 (normalization rules)	リレーショナル・データベース管理システムにおけるデータベース設計の標準規則。
制御ブレイク・レポート (control-break report)	データをグループに分割し、各分割に対する概要情報を生成するレポートまたはデータ表示。分割によって要約データの生成が制御されます。
制限 (restriction)	「選択 (selection)」参照。
整合性制約 (integrity constraints)	create table 文でデータベースの整合性 (database integrity) を記述するモデル。
整合性ルール (integrity rules)	リレーショナル・モデルにおいてデータの正確さと整合性を維持するための規則。「データベースの整合性 (database integrity)」参照。
静的設定パラメータ (static configuration parameter)	値のリセット後に Adaptive Server を再起動する必要がある Adaptive Server 設定パラメータ。「動的設定パラメータ (dynamic configuration parameter)」参照。
精度 (precision)	numeric データ型や decimal データ型で保存できる 10 進数の桁の最大値。精度は、小数点の左右両方のすべての桁が対象になります。
制約 (constraint)	データベース・オブジェクトに適用される規則。これが適用されるデータベース・オブジェクト内のすべてのエントリで、特定の条件を満たすための基準です。たとえば、カラム内のすべての値がユニークになるような制約をカラムに設定することもできます。
セーブポイント (savepoint)	ユーザ定義トランザクション (user-defined transaction) 内にユーザが設定するマーカ。セーブポイントを設定した後、セーブポイント名を指定してロールバック・トランザクション (rollback transaction) コマンドを使用し、セーブポイントまでのコマンドをキャンセルしたり、commit transaction を実行してコマンドを実際に完了することができます。「トランザクション (transaction)」と「ロールバック・トランザクション (rollback transaction)」参照。

セカンダリ・コンパニオン (secondary companion)	フェールオーバー時に、フェールオーバーされたプライマリ Adaptive Server を受け入れるように設定された Adaptive Server 。
セグメント (segment)	特定のデータベースに利用できるデータベース・デバイスの名前付きサブセット。1つ以上のデータベース・デバイスを表すラベルです。セグメントは、特定のデータベース・デバイスにおけるテーブルやインデックスの配置を制御するときを使用します。
セッション (session)	セッションにおいては、ブラウザはサーバとの接続を維持し、複数の要求・応答のサイクルを繰り返すことができます。
セマフォ (semaphore)	セマフォは、現在使用されているデータ構造体に別のタスクがアクセスするのを防ぐ、簡単な内部ロック・メカニズムです。 Adaptive Server では、セマフォを使用して、トランザクション・ログ、ユーザ・ログ・キャッシュ、および I/O デバイスを保護します。 スピンロック (spinlock) と同様に、セマフォは SMP 環境でのみで使用できます。
セマンティック分割 (semantic partitioning)	分割キーの値に基づく、テーブルまたはインデックスの分割。「 ハッシュ分割 (hash partitioning) 」、「 リスト分割 (list partitioning) 」、「 範囲分割 (range partitioning) 」参照。
セミジョイン (semijoin)	ジョイン・アルゴリズムの1つ。基準を満たす最初の内部ローが見つかった時点で各外部ローの内部スキャンを終了します。
セル (cell)	ディストリビューション (distribution) または ヒストグラム (histogram) に含まれる連続した値のペア。各ステップは、ステップの上限値と下限値の間の範囲のカラム値を持つローのセットに対応しています。 ステップ (step) ともいいます。
セルフジョイン (self-join)	テーブル (table) の カラム (column) 内の値を比較するのに使う ジョイン (join) 。この操作ではこのテーブル自身のジョインも行われるので、テーブルにはテンポラリ名を2つ与えるか、 相関名 (correlation names) を与えます。これらの名前は、クエリ後部でカラム名を修飾するのに使用します。
選択 (selection)	テーブル内の ロー (row) のサブセット。制限ともいいます。リレーショナル・システムにおける基本 クエリ (query) 操作の1つです。「 プロデューサ・プロセス (producer process) 」と「 ビュー (view) 」参照。
選択性 (selectivity)	「 インデックスの選択性 (index selectivity) 」と「 ジョイン選択性 (join selectivity) 」参照。
全ページ・ロック・スキーム (allpages-locking scheme)	Adaptive Server の3つのロック・スキームのうちの1つ。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。「 データページ・ロック・スキーム (datapages-locking scheme) 」と「 データロー・ロック・スキーム (datarows-locking scheme) 」参照。

そ

**関連サブクエリ
(correlated
subquery)**

独立して評価できず、その結果が外部クエリによって異なる**サブクエリ (subquery)**。外部クエリで選択される可能性のあるローごとに 1 回実行されるので、**繰り返しサブクエリ (repeating subquery)**とも呼びます。「**ネストされたクエリ (nested queries)**」参照。

**関連のないサブクエリ
(noncorrelated
subquery)**

関連のないサブクエリは、独立したクエリと同じように評価できます。関連のないサブクエリは、ジョインとして記述することも可能で、**Adaptive Server**によりジョインとして処理できます。

**関連名 (correlation
names)**

特に**関連クエリ**や**セルフジョイン (self-join)**などのクエリにおいて、特定のテーブルが果たすさまざまな役割を区別します。関連名は **from** 句で割り当て、テーブル名のあとに指定します。

```
select au1.au_fname, au2.au_fname
from authors au1, authors au2
where au1.zip = au2.zip
```

**早期非アクティブ化
(early deactivation)**

コンポーネントのメソッドが、非アクティブ化するタイミングを指定できるようにします。早期非アクティブ化によって、クライアント・アプリケーションがコンポーネントのインスタンスに関連付けられたリソースによって制限されるのを防ぎ、指定された時間枠でインスタンスがより多くのクライアントに対して機能できるようになります。

総密度 (total density)

カラムまたは一連のカラムに重複が含まれている度合い。複数のヒストグラム・セルにまたがる時もすべて重複と見なされます。「**範囲セル密度 (range cell density)**」参照。

**ソース・テキスト
(source text)**

コンパイル済みオブジェクト (compiled object) を定義する SQL 文。ソース・テキストは、**syscomments** テーブルの **text** カラムに保存されます。更新プロセスで使われるので、削除しないでください。ソース・テキストは **sp_hidetext** を使って暗号化することができます。

ソート順 (sort order)

文字データをソートする順序を決定するために **Adaptive Server** が使う仕様。**照合順 (collating sequence)**ともいいます。

**疎頻度カウント
(sparse frequency
count)**

ドメイン中で、値が連続していない**頻度カウント (frequency count)**。たとえば、整数ドメインで 1、10、100 の値の組は、疎とされますが、1、2、3 は疎とされません。「**密頻度カウント (dense frequency count)**」参照。

**ソフト・フォールト
(soft fault)**

一般に持続性のない **Adaptive Server** の障害。ほとんどのソフト・フォールトは、**dbcc checkstorage** 操作中にユーザがターゲット・データベースを更新したり、データ定義言語 (DDL) コマンドが検出された **dbcc checkstorage** によってターゲット・データベースが一時的に不整合になったときに発生します。これらの障害は、**dbcc checkstorage** を 2 回目に実行するときには起こりません。「**ハード・フォールト (hard fault)**」参照。

存在ジョイン (existence join)	特定のサブクエリの代わりに実行されるジョインの1つ。テーブルを使った通常のネスト反復では一致する値がすべて返されますが、存在ジョインでは最初の値が見つかったと TRUE を返し、処理を終了します。一致する値が見つからない場合は、FALSE を返します。
存在スキャン (existence scan)	スキャン・アルゴリズムの1つ。完全に基準を満たす最初のローが見つかった時点でスキャンを終了します。存在スキャンは、通常、フラット化された exists サブクエリで参照されるテーブルに使用されます。
た	
ターゲット (target)	PowerJ を使用して作成した、 アプリケーション (application) 、アプレット、 クラス (class) 、または集合。
ターゲット・セグメント (target segment)	ソートにおいては、ターゲット・セグメントは、 create index コマンドの完了時に インデックス (index) が保存される セグメント (segment) です。
ターゲット・データベース (target database)	dbcc で検査するデータベース。
ダーティ・バッファ・グラブ (dirty buffer grab)	バッファ・プール用のウォッシュ領域が小さすぎて、クリーン・バッファを必要とするオペレーションが、プールの LRU 側の終端またはビクティム・マーカでダーティ・バッファをディスクに書き込むために I/O 待ちしなければならない場合に発生します。
ダーティ・ページ (dirty pages)	最終書き込み以降に更新されたページ。
ダーティ・リード (dirty read)	最初のトランザクションでローが変更され、最初のトランザクションが変更をコミットする前に、2番目のトランザクションでそのローを読み込むと発生します。最初のトランザクションで変更箇所までロールバックすると、2番目のトランザクションによる読み込みは無効になります。
ターミネータ (terminator)	カラム (column) の区切り文字。
対称型 (symmetrical)	2つの独立した Adaptive Server が互いにフェールオーバー・サーバとして機能する高可用性システム。つまり、各 Adaptive Server がプライマリ・コンパニオンとセカンダリ・コンパニオンの両方として機能します。
対称型マルチプロセッシング (symmetric multiprocessing: SMP)	複数の CPU が同時にクライアント・タスクを処理する環境。すべての CPU であらゆるタスクを実行できます。CPU は、オペレーティング・システムとアプリケーションのすべての機能を対称的に提供および実行します。これは、ノンクラスタードの Adaptive Server 環境です。
対称キー暗号化 (symmetric key encryption)	メッセージの暗号化と復号化に同じアルゴリズム (キー) を使用するプロセス。

ダイナミック・リンク・ライブラリ (dynamic link library: DLL)	アプリケーションにサービスを提供するために Microsoft Windows と IBM OS/2 で使用されるソフトウェア。
タイム・スライス (time slice)	他のタスクにエンジンを自発的に譲り渡す前に、 Adaptive Server エンジン上でタスクを実行できる時間。サーバ上の値を調べるには、次のコマンドを使用します。 <pre>sp_configure time slice</pre>
大容量 I/O (large IO)	複数のデータ・ページの読み込み、書き込みを行う I/O。
代理権限 (proxy authorization)	サーバ内の他のユーザの代行をする (他のユーザになりかわる) 機能。ユーザへの代理権限は、システム・セキュリティ担当者が与えます。代理権限があると、管理者は特定ユーザのパーミッションを調べたり、ユーザのデータベース・オブジェクトを管理することができます。また、多数のユーザの代わりにアプリケーション・サーバがサーバにログインし、プロシージャやコマンドを実行することもできます。
タスク (task)	サービス要求を満たすためにカーネルによってスケジュールされた実行単位。
タプル (tuple)	保管されているテーブルまたは抽出テーブル (table) 内のロー (row)。
タプルのフィルタ (tuple filtering)	入力ストリームが 1 つの実行処理。参照される属性がすべて順番に並んでいることを前提とし、その前提に基づいて、重複するタプル (tuple) を削除します。
多方向ジョイン (multi-way joins)	いくつかのテーブルを含み、それらが 2 つ以上のテーブルにジョインされてスター・ジョイン構成やスノーフレイク・ジョイン構成を形成しているジョイン・クエリ。
探索指数 (search argument)	クエリの where 句内の述部。インデックスを通してローを探すときに使用できます。「SARG」ともいいます。
ダンプ (dump)	データおよびトランザクション・ログ (transaction log) を含むデータベースのバックアップの作成。このアクションは、「データベース・ダンプ」(dump database コマンドを使用) と「トランザクション・ログ・ダンプ」(dump transaction コマンドを使用) の 2 種類のアクションで構成されています。データベース・ダンプには、データベースのすべてのデータおよびトランザクション・ログが含まれ、トランザクション・ログ・ダンプには、データを除くトランザクション・ログが含まれます。トランザクション・ログ・ダンプは、トランザクション・ログからのレコードの削除も実行し、ログ領域を増加します。 また、この用語は、このアクションで得られるデータを指す場合もあります。「ロード (load)」参照。
ダンプ・ストライピング (dump striping)	単一のダンプまたはロード・コマンドに複数のダンプ・デバイスを使用できるようにします。

ダンプ・デバイス (dump device)	データベースやトランザクション・ダンプ用に使う 1 巻のテープ、1 つのパーティション、または 1 つのファイル。ダンプは複数のデバイスにまたがることができ、また多くのダンプを単一のテープ・ボリュームに作成することもできます。
ダンプ・ファイル (dump device)	バックアップ・メディア上で特定のバックアップを識別するために使用されるダンプ・ファイルの名前。この名前は 17 文字以内で指定し、オペレーティング・システムの規則に従う必要があります。名前を入力しない場合は、次の事項に基づいて Backup Server がデフォルト名を生成します。 <ul style="list-style-type: none">• データベース名の最後の 7 文字• 2 桁の西暦年度 (The two-digit year)• 3 桁の年間通し日付 (1 ~ 366) (The three-digit day of the year (1 through 366))• ダンプ・ファイルが作成された時間 (16 進コード) たとえば、 <i>cations93059E100</i> というファイルには、1993 年の 59 日目に作成された <i>publications</i> データベースのコピーが格納されています。
ダンプ・ボリューム (dump volume)	データベースやトランザクション・ダンプ用に使う 1 巻のテープ、1 つのパーティション、または 1 つのファイル。ダンプは複数のボリュームで構成したり、また、複数のダンプを 1 巻のテープ・ボリュームで構成することもできます。
ち	
遅延更新 (deferred update)	2 つの段階で行われる更新オペレーション。最初に、既存のエントリの削除と、新しいエントリの挿入に対するログ・レコードがログに書き込まれます。ただし、データ・ページとインデックスに対する削除変更だけが行われます。次の段階で、ログ・ページが再度スキャンされ、データ・ページとインデックス上で挿入オペレーションが実行されます。「 直接更新 (direct update) 」参照。
チェックポイント (checkpoint)	変更されたページがすべてデータベース・デバイスに確実に書き込まれた時点。
致命的なエラー (fatal errors)	重要度レベルが 19 以上のエラー。ユーザの作業セッションを停止するため、再度ログインが必要です。
抽出テーブル (derived table)	結果が 1 つ以上のカラムの形式になるクエリ式を評価することで、1 つ以上のテーブルから直接または間接的に抽出されたテーブル。抽出テーブルの値は、クエリ式が評価されるときに、基本となるテーブルの値から抽出されます。抽出テーブルはシステム・カタログには記述されず、ディスクにも格納されません。「 SQL 抽出テーブル (SQL derived table) 」と「 抽象プランの抽出テーブル (abstract plan derived table) 」参照。
抽出テーブル式 (derived table expression)	SQL 抽出テーブルを定義するクエリ式。抽出テーブル式は、select、insert、create view の各文の from 句にネストされる select 文で構成されます。

抽出統計 (derived statistics)	検索領域の操作中に計算され、使用される統計。クエリ最適化の間は存続します。抽出統計には、すべての述部 (フィルタリングなど) が適用された後に存在する、変更されたヒストグラム、密度、カラム幅、テーブル統計が含まれます。
抽象プラン (abstract plan)	アクセス・パス上の、Adaptive Server Enterprise のクエリ・プロセッサに対する命令。クエリを実行するためのデータ (ジョイン (join) 順、ジョイン・アルゴリズム、インデックスの使用状況など) の操作に使用します。
抽象プランの抽出テーブル (abstract plan derived table)	クエリの処理、最適化、実行で使用されるテーブル。抽象プランの抽出テーブルは抽象プランの一部であり、エンド・ユーザには表示されない点で、SQL 抽出テーブルと異なります。「抽出テーブル (derived table)」と「SQL 抽出テーブル (SQL derived table)」参照。
調整クエリ・プラン (adjusted query plan)	Adaptive Server は、使用できるワーカー・プロセスに見合うようにクエリ・プランを調整します。調整クエリ・プランは実行時に生成され、使用できるワーカー・プロセスの不足を補います。
直積 (Cartesian product)	ジョインで指定した各テーブルのローの有効な組み合わせ。直積内のロー数は、最初のテーブルのローに 2 番目のテーブルのローを掛けた値になります。直積が設定されると、ジョイン条件を満たさないローは削除されます。
直接更新 (direct update)	1 段階で行われる更新オペレーション。ログ・レコードが書き込まれ、データとインデックス・ページが変更されます。直接更新には、 置き換え更新 (in-place update または update in place) 、 ページ上更新 (on-page update) 、 削除/挿入型直接更新 (delete/insert direct update) の 3 とおりの方法があります。「 遅延更新 (deferred update) 」参照。
直列可能な読み込み (serializable read)	トランザクション内のクエリ (query)。常に同じ一連のローを返します。 幻影ロー (phantom rows) を防ぎます。トランザクション独立性レベル 3 ともいいます。
て	
定義文 (definition statement)	「 ソース・テキスト (source text) 」参照。
定数式 (constant expression)	式を使用するたびに同じ値を返す式。Transact-SQL 構文における文では、定数式には変数やカラム識別子を指定しません。
ディスクの初期化 (disk initialization)	Adaptive Server が使用できるように、 データベース・デバイス (database device) またはファイルを準備するプロセス。初期化したデバイスは、データベースとデータベース・オブジェクトの保管に使用します。データベース・デバイスの初期化は、 ディスクの初期化 (disk initialization) コマンドで行います。
ディスク・ミラーリング (disk mirroring)	重複した Adaptive Server データベース・デバイス。ミラーリング中のデバイスへの書き込みはすべて別の物理デバイスにコピーされ、第 2 のデバイスはミラーリング中のデバイスのコピーそのものになります。デバイス的一方が失敗しても、残りの一方にすべてのトランザクションの最新コピーが保存されます。ディスクのミラーリング・プロセスは、コマンド <code>disk mirror</code> で実行します。

ディスク割り付けの集まり (disk allocation pieces)	ディスク割り付けの集まりは、 Adaptive Server が新しいデータベース・ファイルを構成する基本となる、アロケーション・ユニットの集合です。ディスク割り付けの集まりの最小サイズは、1つの アロケーション・ページ (allocation page) です。
ディストリビューション (distribution)	テーブル・カラムから抽出される、n+1 の値が配列されたセット。すべての値をソートしてから、最初の値とそれに続く (row_count/nth 番目の) 値を選択すると取得されます (row_count はテーブル内のローの数)。
ディストリビューション・ページ (distribution page)	11.9 より前のバージョンの SQL Server と Adaptive Server でデータの分散に関する統計値の格納に使用されていたインデックスと関連付けられたページ。
ディメンション・テーブル (dimension table)	スター・スキーマ内でプライマリ・キーの役割を果たすテーブル。同じスター・スキーマ内のほかのディメンション・テーブルと結合して 複合キー (composite key) を作成し、中央ファクト・テーブルの情報にアクセスしたり、中央ファクト・テーブルの各情報を ジョイン (join) したりできます。
ディレクトリ情報ツリー (directory information tree: DIT)	LDAP サーバのエントリは、エンティティと呼ばれます。各エントリは識別名 (distinguished name: DN) を持ち、階層ツリー構造内に格納されます。このツリー構造をディレクトリ情報ツリーと呼びます。
データ・ウェアハウス (data warehouse)	主に意思決定や解析処理を目的として設計されるサブジェクト指向の情報ストア。データ・ウェアハウスにはマネージャ、市場調査担当者、商業関係者が使用する大量の情報が保存され、市場動向の予測や重要な業務決定を支援します。一般には、データ・ウェアハウスの情報をその利用者が変更することはありません。「 リレーショナル・データベース (relational database) 」参照。
データ・オンリー・ロック (data-only locking)	データページ・ロック・スキーム (datapages-locking scheme) と データロー・ロック・スキーム (datarows-locking scheme) を総称する用語。これらのロック・スキームでは、基本となるデータ・ローまたはデータ・ページだけがロックされ、インデックスロー・ロックとページ・ロックは取得されません。基本となるデータ・ページをロックすることによって、インデックス・ローまたはインデックス・ページが暗黙的にロックされます。
データ型 (datatype)	個々のカラムにおさめる情報の種類とその保存方法を指定します。データ型には、char、int、money などがあります。ユーザは、 Adaptive Server のシステム・データ型に基づいて独自のデータ型を構築できます。
データ型階層 (datatype hierarchy)	さまざまなデータ型の値を使う計算結果を決定する階層。
データ型変換関数 (datatype conversion function)	Adaptive Server では自動的に行われず、あるデータ型から別のデータ型へ変換する式に対して使う関数。
データ・キャッシュ (data cache)	「 バッファ・キャッシュ (buffer cache) 」参照。

データ検索 (data retrieval)	データベースに対するデータ要求動作と結果を受け取る操作。「クエリ (query)」参照。
データ・サーバ・インタフェース (data server interface: DSI)	Replication Server とデータベースの間の接続に対応する Replication Server スレッド。スケジューラ・スレッドと1つまたは複数のエグゼキュータ・スレッドで構成される DSI スレッドは、DSI アウトバウンド・キューからのトランザクションをレプリケート・データ・サーバへ送信します。スケジューラ・スレッドはトランザクションをコミット順にグループ化し、生成されたグループをエグゼキュータ・スレッドへディスパッチします。また、エグゼキュータ・スレッドは関数を関数文字列にマップし、レプリケート・データベース内のトランザクションを実行します。
データ辞書 (data dictionary)	各データベース・オブジェクト (database object) とその構造を記述したシステム・テーブル。
データ修正 (data modification)	insert コマンド、delete コマンド、update コマンドによるデータベース内の情報の追加、削除、変更。
データ整合性 (data integrity)	データベース内のデータの正確さと完全性。
データ定義 (data definition)	データベースを設定し、テーブル、インデックス、ルール、デフォルト、プロシージャ、トリガ、ビューというデータベース・オブジェクトを作成するプロセス。「ソース・テキスト (source text)」参照。
データ定義言語 (data definition language : DDL)	データベースのステータスを取得する Transact-SQL スクリプト。これらのスクリプトは、同じロケーションまたは異なるサーバにデータベースを再作成できます。
データ・フロー・エンジン (data flow engine)	少ない実体化ステップで大規模なプランを処理する実行エンジンを指す用語。
データページのクラスタ率 (datapage-cluster ratio)	ローがインデックス順にアクセスされる時、エクステント I/O を余分に必要としないページ・アクセスの割合。指定されたインデックスを使用するデータ・アクセスに関するエクステント内のデータページの順序と密度は、データベースのクラスタ率によって計測されます。
データページ・ロック・スキーム (datapages-locking scheme)	Adaptive Server の3つのロック・スキームのうちの1つ。データページ・ロックでは、データ・ページだけがロックされます。インデックス・ページ上のページ・ロックは取得されません。「全ページ・ロック・スキーム (allpages-locking scheme)」と「データロー・ロック・スキーム (datarows-locking scheme)」参照。
データベース (database)	「リレーショナル・データベース (relational database)」参照。

データベース・イベント (database event)	データベースのステータスに何らかの変更をもたらすトランザクション。たとえば、トランザクションの開始、ロールバック、コミット、挿入、更新、削除、blob (Java オブジェクト、イメージ、またはテキスト)、データベースを変更させるストアド・プロシージャの実行などです。Replication Adapter と SQL Adapter はデータベース・イベントを検出します。
データベース・オブジェクト (database object)	データベースのコンポーネントの1つ、つまりテーブル、カラム、ビュー、インデックス、プロシージャ、トリガ、デフォルト、またはルールのうちいずれかです。「 オブジェクト (object) 」参照。
データベース・オブジェクト所有者 (database object owner)	データベース・オブジェクト (database object) を作成するユーザ。
データベース管理 (database administration)	データベースの管理、設計、実装、変更、チューニング、拡張に関する業務。「 システム管理 (system administration) 」参照。
データベース管理システム (Database Management System: DBMS)	クエリ処理と現在のタスクのトランザクション管理を行う Adaptive Server の DBMS コンポーネント。たとえば、DBMS は SQL 文を解析、コンパイル、実行して、結果を返します。
データベース所有者 (Database Owner)	データベースの作成者。データベース所有者は、そのデータベース内のデータベース・オブジェクトをすべて制御できます。データベース所有者のログイン名は“dbo”です。
データベース・デバイス (database device)	データベースを構成するオブジェクトを保管する専用の デバイス (device) 。
データベースの整合性 (database integrity)	データベースの有効性と完全性によって証明されるデータベースの特性。データベースの整合性は2つの補完的なコンポーネントで構成されます。これらは、false 情報がすべてデータベースから排除されていることを保証する「有効性」と、true 情報がすべてデータベースに含まれていることを保証する「完全性」です。「 整合性制約 (integrity constraints) 」参照。
データベース・ファイル (database file)	「 テーブル (table) 」参照。
データ・ローのクラスタ率 (datarow cluster ratio)	ローがインデックス順にアクセスされる時、論理 I/O または物理 I/O を余分に必要としないインデックスに対するロー・アクセスの割合。データ・ローのクラスタ率は、データオンリー・ロック・テーブル上のクラスタード・インデックスとノンクラスタード・インデックスごとに管理されます。クラスタード・インデックスでは、各ローが必ずキーの順序になるため、データ・ロー・クラスタ率は常に1 (100% クラスタ化される) です。したがって、クラスタ率は管理されません。

データロー・ロック・スキーム (datarows-locking scheme)	Adaptive Server の3つのロック・スキームのうちの1つ。データロー・ロックでは、データ・ローだけがロックされます。インデックス・ページではロー・ロックは取得されません。「 全ページ・ロック・スキーム (allpages-locking scheme) 」と「 データページ・ロック・スキーム (datapages-locking scheme) 」参照。
テーブル (table)	対応する カラム (column) のある ロー (row) の集まり。論理的にはデータベース・ファイルと同じです。
テーブル・スキャン (table scan)	テーブル内のすべての ロー (row) を読み込むことによって テーブル (table) にアクセスする方法。クエリに条件 (where 句) がいないとき、クエリで指定した句に インデックス (index) がいないとき、 Adaptive Server オプティマイザがテーブル・スキャンよりもインデックスの方がコストがかかるのでインデックスを使うべきでないと判断したときに、テーブル・スキャンを使用します。「 アクセス・メソッド (access method) 」参照。
テーブル正規化ヒストグラム (table-normalized histogram)	update statistics によって計算される ヒストグラム (histogram) 。このヒストグラムのウェイト配列値 (テーブル・ローの割合) の合計は常に 1.0 となります。
テーブルレベル制約 (table-level constraint)	テーブル (table) の複数の カラム (column) に対して値を制限する条件。テーブルレベルの 制約 (constraint) は、create 文にカンマで区切った句として指定します。複数のカラムで動作する制約は、テーブルレベルの制約として宣言します。
テキストおよびイメージ関数 (text and image function)	text データや image データに使う関数。テキスト関数およびイメージ関数には、patindex、textptr、textvalid があります。
テキスト・チェーン (text chain)	テーブル (table) のテキストおよびイメージ値を保存するための特殊なデータ構造。データ・ローには、テキスト・チェーン内のテキストおよびイメージ値のロケーションを表すポインタが保管されています。
デジタル署名 (digital signature)	テキスト／メッセージからユニークな固定長の文字列になった数字を生成する数値アルゴリズムを使用して、デジタル署名は作成されます。この生成された数値はハッシュまたはメッセージ・ダイジェストと呼ばれます。デジタル署名は、不正な変更を検出したり拒否を禁止したりする場合に使用されます。
デッドロック (deadlock)	複数のユーザ・プロセスがそれぞれ別々のページまたはテーブルにロックを設定している状態で、他のプロセスのページまたはテーブルのロックを要求するとデッドロックが発生します。デッドロックが発生すると、CPU 時間の一番少ないトランザクションが強制終了され、その作業はすべてロールバックされます。
デバイス (device)	データベース・オブジェクトを保管するために使われるディスクの一部分またはファイル・システム内のファイル。「 データベース・デバイス (database device) 」参照。
デバイス I/O (device I/O)	データベース・デバイス (database device) に対する読み込みまたは書き込みの動作。

デフォルト (default)	オプションが指定されない場合にシステムが選択するオプション。
デフォルト言語 (default language)	ユーザの場合、ユーザのプロンプトおよびメッセージの表示に使われる言語。set コマンドの <code>sp_modifylogin</code> または <code>language</code> オプションで設定します。 Adaptive Server の場合、ユーザが別の言語を指定しないときに、すべてのユーザに対してプロンプトやメッセージを表示するために使用する言語です。
デフォルト・データベース (default database)	ユーザがログイン時に接続するデータベース。
デマンド・ロック (demand lock)	デマンド・ロックが設定されると、テーブルまたはデータ・ページに追加の共有ロックが設定できなくなります。新しい共有ロック要求を行うには、デマンド・ロック要求が終了するまで待ちます。
転送先アドレス (forwarding address)	ローの新しい位置。転送先アドレスは、ローの元の位置にあります。
転送されたロー (forwarded row)	ローの長さが変わったために、新しいデータ・ページにマイグレートされたデータオンリー・ロック・テーブル内のロー。ローが以前あった位置には、転送されたローの位置へのポインタが格納されます。
テンポラリ・データベース (temporary database)	テンポラリ・テーブルやその他のテンポラリ作業記憶の必要性 (<code>group by</code> や <code>order by</code> の中間結果など) に応じて格納領域を提供するデータベース。 Adaptive Server のテンポラリ・データベースは <code>tempdb</code> です。
と	
統一化ログイン (unified login)	サーバにログインする前にセキュリティ・メカニズムによる認証を受ける機能。この機能によって、ユーザは接続のたびにログイン名とパスワードを指定せずに複数のサーバにログインできます。
等価ジョイン (equijoin)	等価を基にした ジョイン (join) 。
等価パーティション (equipartitioned)	互換性のある分割キーと分割基準を持つ 2 つのテーブル。2 つのテーブルが互換性のあるデータ型の 分割キー (partition key) を同じ数だけ持ち、分割基準 (範囲パーティションのインターバルなど) が同じである場合、2 つのテーブルは等価パーティションであると見なされます。
同時実行性 (concurrency)	独立した、場合によっては競合するプロセスやトランザクションの同時実行 (マルチプロセッシング)。
動的インデックス (dynamic index)	or を使っているクエリを解決するために、 Adaptive Server が構築するワーク・テーブル。基準を満たすローが検索されるたびに、ロー ID がワーク・テーブルに保管されます。ワーク・テーブルは重複のないようにソートされ、ロー ID は値を返すため、再びテーブルに関連付けられます。

動的設定パラメータ (dynamic configuration parameter)	リセットするとただちに更新される Adaptive Server の設定パラメータ。設定パラメータを有効にするために、 Adaptive Server を再起動する必要はありません。「 静的設定パラメータ (static configuration parameter) 」参照。
動的ダンプ (dynamic dump)	データベースがアクティブな間に行われるダンプ。
動的パーティション削除 (dynamic partition elimination)	実行時にテーブルまたはインデックスのパーティションを削除する処理。通常は、 バインドされていない述部 (unbound predicate) を使用する場合に実行されます。
動的負荷分散 (dynamic load distribution)	(クラスタの概念) 論理クラスタ内での負荷のバランスをとるために実行される確立済みのクライアント接続の異なるインスタンスへのマイグレーション。
ドキュメント・タイプ定義 (document type definition: DTD)	XML ドキュメントでの、1つのクラスに特有の要素と属性の定義リストです。
独立性レベル (isolation level)	現在のトランザクションの実行中に禁止される動作の種類を指定します。ANSI 標準では SQL トランザクションの独立性について 4 つのレベルを定義しています。レベル 0 では他のトランザクションが変更されるのを防ぎます。ユーザは set オプション transaction isolation level 、 select または readtext の at isolation 句によって独立性レベルを制御します。レベル 3 は holdlock ですべてのクエリを実行するのと同じです。デフォルトはレベル 1 です。これは「ロック・レベル」とも呼ばれます。
独立並列処理 (independent parallelism)	「 枝分かれの多い並列処理 (bushy parallelism) 」参照。
ドライバ (driver)	外部サービス・プロバイダにインタフェースを提供する Sybase ライブラリ。
トランザクション (transaction)	単一の作業単位として扱われる Transact-SQL 文のグループ。グループ内の文は、すべて実行されるか、まったく実行されないかのどちらかになります。トランザクション中にクエリが実行されたテーブルは、トランザクションが終了するまでロックされます。
トランザクション・オブジェクト (transaction object)	Java クライアント・アプリケーションは、トランザクション・オブジェクトを使用してデータベース接続を確立します。
トランザクション記述子 (transaction descriptor)	Adaptive Server がトランザクションの記述に使用する内部メモリ構造。
トランザクション・コーディネータ (transaction coordinator)	トランザクション・コーディネータは、複数の接続 (ときには複数のデータ・ソース) を含むトランザクションのフローを管理します。

トランザクション属性 (transaction attribute)	コンポーネントがトランザクションにどう参加するかを決定します。
トランザクション独立性レベル (transaction isolation level)	「 独立性レベル (isolation level) 」参照。
トランザクション・モード (transaction mode)	SQL トランザクションで、トランザクションの開始に暗黙的な <code>begin transaction</code> 文を使用するか、しないかを設定。
トランザクション・ログ (transaction log)	データベースに対するすべての変更が記録される システム・テーブル (system table) (syslogs)。
トランザクション・ログのオプション (transaction log options)	トランザクション・ログには、次のようなオプションがあります。 <ul style="list-style-type: none">バックアップ・トランケート・ログ – トランザクション・ログをバックアップしたり、ログの非アクティブな部分を削除したり、バックアップを記録する新しいトランザクション・ログ・エントリを作成したりします。バックアップ・ログ – トランザクション・ログをバックアップしたり、バックアップを記録する新しいトランザクション・ログ・エントリを作成したりします。このオプションは、トランザクション・ログ・エントリを保持します。トランケート・ログ – バックアップを行わずにログの非アクティブな部分を削除したり、ダンプを記録する新しいトランザクション・ログ・エントリを作成したりします。トランケート・オンリー – バックアップを行わずにログの非アクティブな部分を削除し、ダンプを記録する新しいトランザクション・ログ・エントリを作成しません。
トリガ (trigger)	ユーザが指定したテーブルやカラムに <code>insert</code> 、 <code>delete</code> 、 <code>update</code> などの変更コマンドを実行する特殊な形式の ストアド・プロシージャ (stored procedure) 。トリガは、 参照整合性 (referential integrity) を保つときに使用します。
トリガ条件 (trigger condition)	トリガ (trigger) を有効にするための条件。
トリガ・テーブル (trigger table)	トリガ (trigger) が設定された テーブル (table) 。
トリガ・テスト・テーブル (trigger test tables)	データ変更によってキー・カラムに影響が発生すると、 トリガ (trigger) ではトリガ・テスト・テーブルというテンポラリ・ワーク・テーブルを使って新しいカラム値を関連キーと比較します。
トリガ動作 (trigger action)	トリガ (trigger) に指定された動作。

貪欲な検索方式 (greedy search strategy)	クエリ・プランを高速に検索するためにオプティマイザが使用する順列方式。すべてのクエリ・プランを検索するのではなく、非常に粗い基準を使用するため、必ずしも最適な方式ではありません。
な	
内部クエリ (inner query)	「サブクエリ (subquery)」参照。
内部ジョイン (inner join)	制限を満たすジョイン・テーブルのローだけで構成される結果セットを生成するジョイン。ジョイン制限を満たさないローは、ジョイン・テーブルには含まれません。
内部テーブル (inner table)	外部ジョイン (outer join) のテーブル (table) の配置の説明。たとえば、左側のジョインでは、内部テーブル (inner table) は右側のテーブルです。
ナチュラル・ジョイン (natural join)	結び付けられたカラムの値が等価性に基づいて比較され、テーブル内のすべてのカラムが結果に含まれるジョイン (join)。ただし含まれるのは、対応するカラムの組み合わせの内の片方だけです。
名前付きキャッシュ (named cache)	「バッファ・キャッシュ (buffer cache)」参照。
に	
任意アクセス制御 (discretionary access controls: DAC)	オブジェクトへのアクセスが、自分の ID と自分のグループのメンバシップによるアクセスに制限されます。特定のアクセス・パーミッションを持つユーザは、(grant コマンドで付与される) そのパーミッションをほかのユーザに譲渡できるので、制御は任意に行われることになります。「パーミッション (permission)」参照。
ね	
ネイティブの文字セット (native character set)	国際文字セットともいいます。プラットフォーム固有の、言語のサブセットをサポートする文字セット。たとえば、西欧言語などがあります。
ネイティブ・フォーマット (native formats)	-n オプションを使用して作成される、オペレーティング・システム固有のフォーマット。ネイティブ・フォーマットでは通常、凝縮されたオペレーティング・システム・ファイルが作成されます。
ネクストキー・ロック (next-key locking)	幻ロー (phantom rows) の読み込みから繰り返し読み出しトランザクションを保護する方式。
ネストされた select 文 (nested select statements)	「ネストされたクエリ (nested queries)」参照。
ネストされたクエリ (nested queries)	1 つ以上のサブクエリを持つ select 文。

ネットワークとの結び付きのマイグレーション (network affinity migration)	あるエンジンから別のエンジンにネットワーク I/O を移動するプロセス。このマイグレーションをサポートする SMP システムでは、 Adaptive Server が、そのすべてのエンジン間にネットワーク I/O の負荷を分散させることができます。
の	
ノード (node)	高可用性システムのマシン。クラスタでは、 Adaptive Server インスタンスをホストするマシン (ハードウェア)。
ノーマル・コンパニオン・モード (normal companion mode)	高可用性システムの 2 つの Adaptive Server が独立したサーバとして機能し、定期保守やシステム障害時にフェールオーバーされるように設定されているモード。
ノンクラスタード・インデックス (nonclustered index)	キー値と、データへのポインタを保存する インデックス (index) 。 リーフ・レベル (leaf level) はデータそのものは含まず、データ・ページを指します。
は	
パーティション (partition)	テーブルおよびインデックスの一部を格納し、独立して管理できる、ユーザ作成のデータベース・オブジェクト。 テーブル (table) や インデックス (index) を、個別の物理デバイスに常駐する複数の パーティション (partition) に分割することにより、同時挿入のパフォーマンスが向上し、並列クエリとソートが処理しやすくなります。テーブル・ローを複数のパーティションに分割する方式には、ラウンドロビン分割、リスト分割、範囲分割、ハッシュ分割の 4 種類があります。
パーティション除去 (partition elimination)	分割キーに関する 述部 (predicate) を持つクエリでは、パーティション削除によって、特定の述部について参照されないパーティションを調べることができます。ただし、現在パーティション削除に使用できる述部は、column <relop> <literal> のフォーマットの 1 つのテーブル上で、結合述部 (AND キーワードで接続された SQL クエリ内の句) または非結合述部 (OR キーワードで接続された SQL クエリ内の句) として限定する必要があります。
ハード・フォールト (hard fault)	dbcc checkstorage の操作中に Adaptive Server が検出した持続的な障害。ハード・フォールトは、 Adaptive Server を再起動しても修復できません。「 ソフト・フォールト (soft fault) 」参照。
パーミッション (permission)	特定の動作を特定のデータベース・オブジェクトに対して行ったり、特定のコマンドを実行する権限。
排除 (pruning)	オプティマイザが最適な実行プランを検索するテクニック。最適な全体プランの一部にできるサブプランだけを保持します。オプティマイザは、コストベースの排除とヒューリスティックベースの排除を使用します。
排他ロー・ロック (exclusive row lock)	書き込みオペレーション中に、データ・ロー上で取得されるロックのタイプ。ほかのトランザクションは、そのロー上で排他ロック、更新ロック、または共有ロックを取得できなくなります。

排他ロック (exclusive lock)	書き込み処理中にオブジェクトに設定されるロックの 1 つ。他のトランザクションによる排他ロック、更新ロック、または共有ロックはできません。排他ロックはテーブルやページに設定されます。
パイプライン処理 (pipelining)	2 組のスレッドが、それぞれプロデューサとコンシューマとして動作する処理。プロデューサはデータを共有バッファに格納し、コンシューマは、共有バッファ内のデータをプロデューサと同時に処理できます。
パイプライン並列処理 (pipelined parallelism)	複数のステップから成る SQL オペレーションにおいて、独立した各ステップが、その前のステップの完了前に実行を開始できる処理方式。1 つのクエリを複数のプロセッサで処理できるため、応答時間を短縮できます。
バインド (binding)	デフォルト (またはルール) とテーブル・カラム (またはユーザ・データ型) の関連付け。ルール (またはデフォルト) がテーブル・カラム (または指定されたユーザ・データ型) にバインドされると、カラムに入力できるデータに影響します。バインドは、名前付きデータ・キャッシュとデータベース、テーブル、またはインデックスの関連付けを指す場合もあります。データベース、テーブル、またはインデックスが名前付きデータ・キャッシュにバインドされると、データベース、テーブル、またはインデックスから読み込まれたデータは名前付きキャッシュに移動します。
バインドされていない述部 (unbound predicate)	ローカル変数など、コンパイル時に定数が不明な述部。
パススルー・モード (passthrough mode)	ネイティブな構文で、クライアントがリモート・サーバと通信できるようにするモード。このモードでは Transact-SQL パーサとコンパイラはスキップされ、ユーザから受け取った各言語「バッチ」は、ユーザが接続されたサーバに直接渡されます。
パスワードの暗号化 (password encryption)	解読できない暗号化形式でパスワードを保存するプロセス。
バックアップ (backup)	データベースやトランザクション・ログのコピー。メディア障害が発生したときのリカバリに使用します。
パッケージ (package)	パッケージは、アプリケーションのビジネス・ロジックのサービスや機能を提供するために連動するコンポーネントのコレクションです。パッケージは、trust の境界を定義し、その定義内でコンポーネントは容易に通信します。それぞれのパッケージは、配布の単位として動作し、アプリケーション・リソースをグループ化して配備および管理を容易にします。
ハッシュ・キー (hash key)	管理するオブジェクトに対して、ハッシュ関数で生成される値。
ハッシュ・テーブル (hash table)	同じハッシュ・キー (hash key) を持つオブジェクトに高速にアクセスするハッシュ・アルゴリズムのサポートに使用します。

ハッシュ・バケット (hash buckets)	ハッシュ・テーブル (hash table) は、ハッシュ・バケットで構成されています。各ハッシュ・バケットはハッシュ・キー (hash key) によって選択されます。
ハッシュ分割 (hash partitioning)	分割キー (partition key) の値に基づくテーブル (table) またはインデックス (index) の分割。システム定義のハッシュ関数によって分割キーが処理され、各ロー (row) が属するパーティション (partition) が決定されます。
ハッシュベースの集約 (hash-based aggregation)	group by 集合を評価する方式の 1 つ。グループ化カラムのハッシュ・キーによってグループを検索します。
バッチ (batch)	バッチの終わりを示すシグナルで終了する 1 つ以上の Transact-SQL 文。バッチの終わりを示すシグナルによって、Transact-SQL 文は処理のために Adaptive Server に送られます。
バッファ (buffer)	メモリ・プール内の格納単位。1 つのバッファ・キャッシュ (buffer cache) で、さまざまな I/O サイズつまりバッファ・サイズに設定したプールに対応できます。プール内のバッファはすべて同じサイズです。プールが 16K I/O に設定されている場合、すべてのバッファは 8 データ・ページを保持する 16K となります。バッファは 1 つの単位として扱われます。つまり、バッファ内の各データ・ページに対する読み込み、書き込み、フラッシュは、キャッシュから、すべて同時に行われます。
バッファ・キャッシュ (buffer cache)	Adaptive Server 内のメモリ領域。ページの管理に必要なデータ構造以外に、データベース・ページのメモリ内イメージが含まれます。各キャッシュには設定用のユニークな名前が付けられます。デフォルトでは、Adaptive Server は「デフォルト・データ・キャッシュ」という 1 つのキャッシュを備えています。ユーザが設定するキャッシュを「ユーザ定義キャッシュ」、バッファ・キャッシュをデータ・キャッシュ、または名前付きキャッシュと呼びます。
バッファ再利用方式 (buffer reuse strategy)	同じバッファがすぐに再利用できるように、データ・キャッシュのキャッシュ・チェーンの LRU 側の終端にページを読み込むこと。この方式では、大量のページ読み込みが必要な select コマンドによって、キャッシュからほかのデータがフラッシュされるのを制限します。
バッファ置換方式 (buffer replacement strategy)	「バッファ再利用方式 (buffer reuse strategy)」参照。
バッファ・プール (buffer pool)	バッファ・キャッシュ (buffer cache) 内のメモリ領域。MRU/LRU (最も最近に使用された／最も長い間使用されていない) リストにリンクされたバッファ・セットをおさめます。
パフォーマンス (performance)	Adaptive Server がクエリを実行し、結果を返す速度。テーブルのインデックス、ロー・パーティションとファイルのどちらを使用するか、またはセグメントなど、パフォーマンスはいくつかの要素の影響を受けます。
パブリック・キー暗号法 (public-key cryptography)	インターネット上で機密データを安全に送信するために使われる暗号法。複数のメカニズム (暗号化、キー交換、デジタル署名、デジタル証明書など) で構成されます。

パブリック・キーとプライベート・キー (public/private keys)	非対称キーともいいます。パブリック・キーとプライベート・キーはペアになっており、1つがパブリック・キー、もう1つがプライベート・キーであり、メッセージの暗号化と復号化に使用されます。パブリック・キーとプライベート・キーは、パブリック・キー暗号法で使用されています。
パラメータ (parameter)	ストアド・プロシージャ (stored procedure) またはシステム・プロシージャ (system procedure) に対する引数 (argument)。
バルク・コピー (bulk copy)	データベースに対するデータの出入力コピー・プロセス。Adaptive Server ではこの操作を bcp ユーティリティで行います。
ハロウィーン問題 (Halloween problem)	カーソル (cursor) の更新にともなう異常。結果セットに1つのロー (row) が2度現われたように見えます。クライアントでインデックス・キーを更新し、更新したインデックス・ローが結果セット中で下に移動すると発生します。
範囲クエリ (range query)	特定範囲の値のデータを要求するクエリ (query)。大なり記号や小なり記号を使ったクエリ、between を使ったクエリ、like を使ったクエリがあります。
範囲セル (range cell)	ヒストグラム (histogram) のセル。値の範囲を表します。セルのウェイトは、セルの下限値より大きく、上限値より小さいか上限値と同等なカラム値の割合を表します。「頻度セル (frequency cell)」参照。
範囲セル密度 (range cell density)	カラムまたは一連のカラムに重複した値が含まれている割合。重複した値が複数の分布にまたがる場合は、重複した値の一部が削除されます。「総密度 (total density)」参照。
範囲分割 (range partitioning)	1つまたは複数の属性のデータ・セットを、値の範囲に基づいて分割します。したがって、各ローを特定のパーティションに確実に格納できます。
範囲ロック (range lock)	トランザクション独立性レベル3の範囲クエリによって取得されたロック。幻ローを防ぐ目的で使用されます。範囲ロック中は、繰り返し読み出しトランザクションが完了するまで、他のユーザによる範囲への挿入が禁止されます。範囲ロックは、データオンリーのロック・テーブルで使用されます。
反復子 (iterator)	実行エンジンの処理の1つ。クエリ結果は反復子を使用してカプセル化されます。反復子は独立したソフトウェア・オブジェクトで、データ・マネージャ (インデックスやテーブル・スキャン) やほかの反復子からローのストリームを受け取り、クライアントやほかの反復子が要求できるローのストリームを生成します。反復子の役割は、多数のノードにまたがる多数のデータ・セットの反復を逐次または並列で処理することです。反復子が外部反復子 (別の反復子) である場合、反復子はデータ・ストリームのソースを認識できません。内部反復子 (反復子自体が生成した反復子) である場合は、ソースを認識できます。反復子は、データ・セットが反復されるたびに、その反復子の指定に従ってデータを操作し、事前定義された動作を処理対象のデータ・セットに適用します。たとえば、ディスク上のテーブルからローをスキャンする動作は、反復子による動作の一種です。反復子の重要な機能は、反復子の種類とそれに関連付けられている動作には関係なく、すべての反復子が同じ手順に従い、同じ外部インタフェースを持つという点です。すべての反復子は、データ・ストリームを開き、ストリームの読み取りを繰り返し実行し、データを処理し、ストリームを閉じます。

汎用インデックス・キー (generalized index key)	ユーザ定義の汎用関数の呼び出しによって生成されるインデックス・キー。
汎用カラム (generic column)	通常はクエリ (query) で参照されるテーブル (table) のカラム (column) を指しますが、式のジョインに使用できる式 (expression) などを抽象的に指す場合もあります。
汎用テーブル (generic table)	通常はクエリ (query) で参照されるテーブル (table) を指しますが、オプティマイザがジョイン順に配置するオブジェクトを抽象的に指す場合もあります。たとえば、サブクエリは汎用テーブルの一例です。
ひ	
ヒープ・テーブル (heap table)	クラスタード・インデックス (clustered index) のないテーブル。データはヒープ構造で保存されます。データ・ローの保存順序には特に指定はなく、新しいデータは常にページ・チェーンの最後に挿入されます。典型的なヒープ・テーブル (heap table) には、クラスタード・インデックスのない非分割テーブルがありますが、クラスタード・インデックスのない分割テーブルもヒープ・テーブルとして分類しています。
ヒストグラム・チューニング係数 (histogram tuning factor)	頻度セル (frequency cell) が存在する場合にのみ、デフォルトのステップ数または指定されたステップ数を超えるステップ数をヒストグラム (histogram) で設定するために使用するチューニング係数。たとえば、ディストリビューションに頻度セル (frequency cell) が存在し、この係数が 3 の場合、ステップ数をデフォルトの 20 から 60 に増やすことができます。
比較演算子 (comparison operator)	クエリ内で 2 つの値を比較する記号。比較演算子には等号 (=)、より大きい (>)、より小さい (<)、以上 (>=)、以下 (<=)、不等 (!=)、より大きくない (!>)、より小さくない (!<) があります。「算術演算子 (arithmetic operator)」参照。
引数 (argument)	関数または関数を評価するのに必要なプロシージャに指定する値。
ビジネス・イベント (business event)	株価変更の転送など、特定のビジネス・オペレーションによって生成されるイベント。これに対し、ユーザ・ログインはビジネス・イベントではありません。
ビジネス・ルール (business rule)	入力を制限し、実社会の要求に基づいて更新する規則。たとえば、sales データベースにおいて、在庫不足の品目の販売を禁止する条件を設定したとします。Adaptive Server では、sales_orders テーブルに追加する前にローを確認し、ビジネス・ルールに従って要件を quantity カラムに適用するといった処理ができます。
ヒステレシス (hysteresis)	セグメント (segment) 上のスレッショルドの間隔を制御し、スレッショルド (threshold) に関連するストアド・プロシージャ (stored procedure) が頻繁にトリガされすぎないようにするために使用される値。

ヒストグラム (histogram)	それぞれのセル (cell) がウェイトを持つセルのセット。各セルには上限、下限、範囲内のローの割合を表す 0 ~ 1 の間の浮動小数値があります。カラム値の Adaptive Server 統計は、sysstatistics システム・テーブルにヒストグラム (histogram) として保存されます。これらの統計はクエリ・オプティマイザで使用されます。
ヒストグラム・ウェイト配列 (histogram weight array)	ヒストグラム (histogram) に関する float 値の配列。セル (cell) で選択されたテーブル (table) の割合 (テーブル正規化ヒストグラムの場合) か、特定の述部で選択されたセルの割合 (セル正規化ヒストグラムの場合) のいずれかを表します。
非対称型 (asymmetrical)	1つのプライマリ・コンパニオンと1つのセカンダリ・コンパニオンで構成される高可用性システム。非対称型システムでは、プライマリ・コンパニオンだけをフェールオーバーできます。このシステムでは、セカンダリ Adaptive Server を「ホット・スタンバイ」ともいいます。
左側の深いツリー・プラン (left deep tree plan)	テーブルのセットの「ジョイン順」を「ツリーベース」で言い換えた語。右側のノードが常にリーフ・ノードである、ツリー状のクエリ・プラン構造を指します。このツリー内でのテーブルの順序は、set forceplan オプションの影響を受けます。
日付関数 (date function)	日付や時刻の情報を表示したり、日付や時刻の値を操作する関数。日付関数には、getdate、datename、datepart、datediff、dateadd があります。
日付部 (date part)	日、月、年など、Transact-SQL 日付関数で認識できる日付の一部分。
否認防止 (non-repudiation)	ドキュメントの有効性の取り消しの禁止。インターネット接続については、デジタル署名が否認防止を提供します。送信者は、送信したメッセージを取り消すことができません。
非マッチング・インデックス・スキャン (nonmatching index scan)	プレフィクス・サブセット (prefix subset) の一部に一致が見られるものの、探索指数がインデックス・キー・カラムの複合キー (composite key) にならない場合の、ノンクラスタード・インデックスを使ったスキャン。スキャンは最低キー値から最高キー値へのインデックスを使って行われ、クエリで指定した一致箇所を検索します。この種のスキャンは、クエリで参照されるテーブルのカラムがすべてインデックスに含まれている場合に、ノンクラスタード・インデックスで行われます。非マッチング・インデックス・スキャンは、テーブル・スキャンよりコストはかかりませんが、マッチング・インデックス・スキャン (matching index scan) よりもかかります。
ビュー (view)	データベースにオブジェクトとして保存される名前付き select 文 (statement)。1つ以上のテーブルからローやカラムのサブセットを表示するとき使用します。「プロデューサ・プロセス (producer process)」と「選択 (selection)」参照。
ビューの解析 (view resolution)	ビュー (view) に関連するクエリにおいて、クエリ (query) 内のデータベース・オブジェクト (database object) の有効性を検証し、クエリとビューの保存されている定義を結合する処理。

ヒューリスティックベースの排除 (heuristic-based pruning)	クエリに適用できる前もって決定された一連の規則に基づいて、検索領域の一部 (特定のツリーの形や順列など) を省略する最適化テクニック。
評価済み設定 (evaluated configuration)	NSA (National Security Agency) が 1996 年に HP 9000 HP-UX BLS, 9.09+ プラットフォームによって、C2 セキュリティ・レベルで評価した SQL Server の設定。リモート・プロシージャやシステム・テーブルへの直接更新などの SQL Server の一部の機能は、この評価済み設定から除外されています。評価済み設定から除外されたすべての機能のリストについては、『SQL Server Installation and Configuration Guide for HP 9000 HP-UX BLS, 9.09+』の付録 A を参照してください。
表示精度 (display precision)	real 値および float 値の場合に、デフォルト表示フォーマットで得られる有効なバイナリ桁数。内部的には、real 値と float 値は、構築されたプラットフォーム固有のデータ型以下の精度で保管されます。Sybase における real 値の表示用の精度は 9 桁、float 値は 17 桁です。
非連鎖トランザクション・モード (unchained transaction mode)	トランザクションを完了するために commit transaction や rollback transaction 文と対になる明示的 begin transaction 文が必要なトランザクション (transaction) モード。「連鎖トランザクション・モード (chained transaction mode)」参照。
頻度カウント (frequency count)	ドメイン内で、少ない数の値でデータ分散をモデリングする方法。たとえば、gender カラムは “M” および “F” という 2 つの値でモデル化できます。これらの値のカウントは、それぞれおよそ 0.48、0.52 です。
頻度セル (frequency cell)	頻度カウント (frequency count) を表すヒストグラム (histogram) 内のセル。頻度セルのウェイトは、そのセルの値と一致する値を持つカラムの割合を示します。「範囲セル (range cell)」参照。
ふ	
ファクト・テーブル (fact table)	スター・データベース・スキーマのメイン・テーブル。複数のディメンション・テーブルの外部キーである属性で構成される、複合キーを持ちます。
ファミリー (family)	コーディネーティング・プロセスは、ワーカー・プロセスのプールから 4 つのワーカー・プロセスを要求する。コーディネーティング・プロセス (coordinating process) は、ワーカー・プロセス (worker process) とセットでファミリーと呼ばれます。
フィールド (field)	「カラム (column)」参照。
フィルタ (filter)	ウィンドウまたはダイアログ内で表示／非表示にするオブジェクトを指定するオペレーション。
ブール式 (Boolean expression)	TRUE (1) か FALSE (0) を判定する式。ブール式は、一般には if 条件や while 条件などのフロー文の制御に使用します。

フェールオーバー (failover)	フェールオーバー時には Adaptive Server は別のマシンにマイグレートし、そのマシンがフェールオーバーされた Adaptive Server の管理を引き受けます。定期保守や、 Adaptive Server または Adaptive Server を実行するマシンの障害によって、フェールオーバーが発生する可能性があります。
フェールオーバー・インスタンス (failover instance)	(クラスタの概念) 論理クラスタの1つまたは複数のベース・インスタンスが実行できない場合にその論理クラスタを実行できるインスタンス。
フェールオーバー・グループ (failover group)	(クラスタの概念) 論理クラスタに対して定義されている、一連のフェールオーバー・インスタンス。フェールオーバー・グループを使用すると、フェールオーバー・インスタンスの優先設定および順位を指定できます。
フェールオーバー・モード (failover mode)	プライマリ・コンパニオンがフェールオーバーされてセカンダリ・コンパニオン上で実行するようになった後の、プライマリ・コンパニオンのモード。
フェールバック (failback)	Adaptive Server が元のマシンに戻って再起動する、予定されたイベント。このイベントでは、フェールオーバーされたデータベース、デバイス、クライアント接続が、セカンダリ・コンパニオンから再起動したプライマリ・コンパニオンへ移動します。
フェッチ (fetch)	フェッチは、1つ以上のローを取得し、現在のカーソル位置をカーソル結果セットに移動します。カーソル・フェッチとも呼びます。
フォーマット・ファイル (format file)	Adaptive Server データベースのテーブルからオペレーティング・システム・ファイルにデータをコピーするために、 bcp が作成するファイル。フォーマット・ファイルにはコピーするデータのフォーマット方法に関する情報が含まれており、 Adaptive Server テーブルにデータをコピーする場合や、さらにコピーを追加する場合に使用できます。
フォーム (form)	ユーザがボタン、テキスト・ボックスなど、ユーザ・インタフェースのその他の要素を配置するアプリケーション・ウィンドウ。
負荷管理 (workload manager)	(クラスタの概念) リソース割り当て、可用性、および負荷分散のアプリケーションレベルでの管理を提供する Adaptive Server モジュール。
負荷スコア (load score)	(クラスタの概念) 1つのインスタンスの総負荷を計算した値。1つの論理クラスタ内の異なるインスタンス間の相対負荷または1つのインスタンスの異なるタイミング間での相対負荷を比較するために使用できる単位のない数値。
負荷プロファイル (load profile)	(クラスタの概念) 論理クラスタ内でインスタンスへの相対負荷を決定するために使用される重み付けされた測定基準のセット。独自の負荷プロファイルを作成することも、 Sybase によって用意されているプロファイルのいずれかを使用することもできます。
複合インデックス (omposite indexes)	1つ以上のカラムを持つインデックス。複合インデックスは、論理的関係のため、2つ以上のカラムを単位として検索する必要があるときに使用します。
複合キー (composite key)	複数のカラムを含むインデックス・キー。たとえば、authors (aulname、aufname) などがあります。

複製 (replication)	データベースにおいて、1つのデータベースのデータの変更(レコードの作成、更新、削除を含む)が、他のデータベースの対応するレコードにも適用されるプロセス。
物理 I/O コスト (physical I/O cost)	オプティマイザが実行する物理読み込みの回数の見積もり。
物理演算子 (physical operator)	物理演算子の実装アルゴリズム。たとえば、論理演算子“join”は、物理演算子アルゴリズム“merge join”、“hash join”、“nested loop join”で実装されます。
物理キー (physical key)	テーブルにインデックスを定義するために create index 文内で使用する1つまたは複数のカラム名。テーブルの物理キーは、 論理キー (logical key) と必ずしも同じである必要はありません。
物理クラスタ (physical cluster)	(クラスタの概念) 特定のクォーラム・ディスク、メンバー・インスタンス、および相互接続情報のある共有ディスク・クラスタ。物理クラスタ内のすべてのインスタンスは、データベースの1つのインストールへの直接アクセスを持ち、クラスタのメンバシップ・サービスによってモニタおよび管理されます。
物理設計 (physical design)	サーバ上に実際にデータベースを作成する Transact-SQL データ定義コマンドに対する 論理設計 (logical design) のマッピング。
物理プロパティ (physical property)	物理演算子に関連付けられたプロパティ。その演算子で実装されている実際のアルゴリズムと、その子の物理プロパティに依存します(したがって、再帰的に、サブプラン内の物理演算子に依存します)。たとえば、通常は、インデックス・スキャンやソートによって返される外部の子の順序が、後続のジョイン演算子の評価後に継承されます。しかし、サブプランで使用されている基本となる演算子によっては、同等クラス内の各プランが異なる順序を持つ可能性もあります。
物理読み込み (physical read)	データ、インデックス、またはログ・ページにアクセスするためのディスク I/O。 Adaptive Server は、クエリを最適化するとき、物理読み込みと論理読み込みを見積もります。「 論理読み込み (logical read) 」参照。
不等価ジョイン (not-equal join)	不等号にもとづくジョイン。
不完全な文字 (Partial character)	マルチバイト文字セットを使用する文字列データにおけるマルチバイト文字の不完全な(トランケートされた)連続バイト。通常は、文字値の末尾に現れます。
付与オプション付き (with grant option)	パーミッションを付与するときのオプション。指定されたユーザまたはグループが他のユーザやグループに同一のパーミッションを付与できます。
プライマリ・キー (primary key)	テーブル内の ロー (row) をユニークに識別する値を持つ1つまたは複数の カラム (column) 。
プライマリ・コンパニオン (primary companion)	Adaptive Server の1つ。フェールオーバー時には、この Adaptive Server のデータベースと接続がセカンダリ Adaptive Server にマイグレートされます。

プラグイン (plug-in)	基本アプリケーションの機能を強化するソフトウェア。
フラグメント (fragment)	create や alter database で、デバイス上に領域の一部だけを割り付けたとき、その部分をフラグメントと呼びます。
プラットフォーム (platform)	アプリケーション・プログラムを動作させる基盤となるコンピュータ・システム。たとえば、HPUX と Solaris はいずれもプラットフォームです。プラットフォームは、オペレーティング・システム (プラットフォーム上で動作するソフトウェア) とは異なります。
プラン・キャッシュ (plan cache)	オプティマイザが、後で完全なプランを作成するときに役立つ可能性がある部分プラン (プランのフラグメント) をプロシージャ・キャッシュに格納すること。プラン・キャッシュはクエリのコンパイル中にだけ存在し、クエリの実行前に解放されます。
ブランチ・プロセッサ (branch processor)	切り替え論理を含むルート上のポイント。単一のイベントはブランチ・プロセッサに入力可能であり、複数のイベントはサブブランチに分けられます。一般的なブランチ用アプリケーションは、メッセージのコピーを複数のサブブランチにファンアウトします。
フリーロック・リスト (freelock list)	複数のエンジンからなる Adaptive Server では、エンジンごとに専用のフリーロック・リスト (ロック要求に対応するために利用できるロックのリスト) があります。エンジン・フリーロックでロックがなくなると、Adaptive Server はグローバル・フリーロック・リストからエンジン・フリーロック・リストにリストを移動します。
プリフェッチ (prefetch)	マルチページ I/O をテーブル、ノンクラスタード・インデックス、またはトランザクション・ログに実行するプロセス。ログを取る場合、サーバは最高 256 ページをフェッチできます。ログを取らないテーブルやインデックスでは、サーバは最高で 8 ページをフェッチできます。
プリフェッチ方式 (prefetch strategy)	「プリフェッチ (prefetch)」参照。
プレフィクス・サブセット (prefix subset)	複合インデックス内のキーの参照に使用します。インデックスの先頭のカラムを指定すると、検索値はプレフィクス・サブセットを構成します。カラム A、B、C のインデックスの場合、プレフィクス・サブセットは A、AB、ABC となります。ただし、AC、B、BC、C とはなりません。「マッチング・インデックス・スキャン (matching index scan)」と「非マッチング・インデックス・スキャン (nonmatching index scan)」参照。
フロー制御言語 (control-of-flow language)	Transact-SQL 文の実行フローを制御する、Transact-SQL のプログラミング的構造 (if、else、while、goto など)。
プロキシ・データベース (proxy databases)	プライマリ・コンパニオン上の各ユーザ・データベースごとにセカンダリ・コンパニオン上に作成されるプレース・ホルダ・データベース。プロキシ・データベースはデータベース名を予約して、フェールオーバー時にシステム上ですべてのデータベース名がユニークであるようにします。

プロキシ・テーブル (proxy table)	リモート・サーバ上のテーブルにマップされているローカル・テーブル。プロキシ・テーブルにはメタデータが含まれており、リモート・テーブルをローカル・テーブル同様にアクセスするために使用します。
プロシージャ (procedure)	ある名前で作成された、SQL 文およびオプションのフロー制御文の集合。Adaptive Server が提供するプロシージャは、システム・プロシージャと呼ばれます。
プロシージャ・キャッシュ (procedure cache)	ストアド・プロシージャ、バッチ・クエリ・プラン、トリガ、ステートメント・キャッシュ、データ変更追跡に使用されたり、クエリのコンパイルや統計の更新などの、短期的にメモリを必要とするさまざまなものに使用される長期メモリ。
プロセス (process)	オペレーティング・システムによって物理 CPU 上にスケジュール調整されている実行環境。
プロセスとの結び付き (process affinity)	特定の Adaptive Server タスクを特定のエンジン上で実行したり、特定のエンジンを特定の CPU で実行するプロセス。
ブロック (blocking)	ロックによる待機。別のプロセスが互換性のないロックを保持する場合、ロー、ページ、またはテーブル上でロックを必要とするタスクは、待機またはブロックする必要があります。
プロデューサ・プロセス (producer process)	並列ソート (parallel sort) では、プロデューサ・プロセスは入力テーブルからデータを同時に読み込み、各データが属する範囲を判断して、適正な範囲に対応したコンシューマ・プロセスにデータ値を分配します。
文 (statement)	1 つ以上の完全な Transact-SQL コマンド (command)。文は、一般に select などのキーワード (keyword) と、それに続く from や where などの句から構成されています。
分割キー (partition key)	各テーブル・ローのパーティション割り当てを決定する、ユーザ定義の分割カラムの値。セマンティック分割方式 (ハッシュ分割 (hash partitioning)、リスト分割 (list partitioning)、範囲分割 (range partitioning)) でのみ使用されます。
分割キー (partitioning key)	パーティション仕様を評価する検索条件。この重要な仕様の指定に関与する一連のカラムを分割キーと呼びます。
分割スキュー (partition skew)	複数のパーティション (partition) にわたってデータが不均等に分布していること。
分割並列処理 (partitioned parallelism)	データを複数の物理パーティションに分割して、各パーティションに並列でアクセスし、作業スレッドで各パーティションを管理できるようにします。このような構成の IO 並列処理と CPU 並列処理では、パーティション (partition) の数に比例して SQL クエリの処理を高速化できます。
文ブロック (statement block)	キーワード (keyword) begin と end の間にエンクローズされ、1 つの単位として扱われる一連の Transact-SQL 文 (statement)。

分類 (classification)	セキュリティの階層レベル。たとえば Top Secret は、Secret よりも高い分類値になります。
へ	
並列クエリ・オプティマイザ (parallel query optimizer)	オプティマイザのコンポーネントの1つ。プランにスケジューリング情報を追加し、プランを再評価し、最適な並列プランを生成します。並列クエリ・オプティマイザは、リソース使用状況の注釈が付けられている一連の並列プランを検討するスケジューラです。クエリに使用できるリソースの合計に基づいて、応答時間に応じた最適なプランを決定します。
並列処理 (parallel processing)	Adaptive Server タスクやサブタスクを同時に実行すること。複数のプロセスでメモリを共有したり、いくつかの種類の同期メソッドを使って相互にメッセージを交換します。
並列ソート (parallel sort)	複数のワーカー・プロセスが並列にデータをソートする技術。create index コマンド、コマンドまたは内部ソートを必要とするクエリで使用されます。単一プロセスでは、入力データが別々の範囲に分割されます。複数のプロセスでは、各範囲のデータ・ローがソートされ、サブインデックスが作成されます。次に、単一プロセスによってソートされた範囲とインデックスが、1つのソート済みインデックスまたは結果セットに結合されます。
並列度 (degree of parallelism)	オプティマイザがクエリを並列で実行するときを選択するワーカー・プロセスの数。並列度はクエリに対する並列度の上限値と、オプティマイザが指定する並列度のレベルの両方によって決まります。
ページ (page)	ディスクに対して読み込みまたは書き込みのできる最小単位である 2K、4K、8K、または 16K ブロックのデータ。
ページ上更新 (on-page update)	直接更新 (direct update) 操作の1つ。データ・ローの長さが変化するときに行われます。変更されたデータ・ローは同じデータ・ページに残りますが、そのページ上の他のローは移動する可能性があります。「置き換え更新 (in-place update または update in place)」と「削除／挿入型直接更新 (delete/insert direct update)」参照。
ページ分割 (page split)	新しいデータやインデックス・ローをページ (page) に追加するとき、新しいローをおさめるだけの領域がないときに、 Adaptive Server が行う動作。通常は既存のページのデータが、新しく割り付けるページと既存のページの両方がほぼ等分になるように分割されます。
ページ・ロック (page lock)	ページ・ロックは、データまたはインデックス・ページ全体をロックします。 Adaptive Server は、データに対するユーザ間の競合を減らし、同時実行性を向上させるため、できる限りページ・ロックを使用します。ページ・ロックはテーブル・ロックよりも制限が少なくなります。
ベース・インスタンス (base instance)	(クラスタの概念) 論理クラスタに割り当てられたインスタンス。通常の場合、このインスタンス上で論理クラスタが実行されます。

ベース・カラム (base column)	計算カラム・インデックスまたは機能ベースのインデックスを定義する基本となるカラム。
ベース・テーブル (base tables)	ビューの基本となる永久テーブル。基本となるテーブルともいいます。
ベクトル集合 (vector aggregate)	group by 句のある 集合関数 (aggregate function) を使用した結果として得られる値。「 スカラ集合 (scalar aggregate) 」参照。
変換 (conversion)	「 文字セット変換 (character set conversion) 」参照。
変数 (variable)	値を割り当てられている エンティティ (entity) 。Adaptive Server には、 ローカル変数 (local variable) と グローバル変数 (global variable) という 2 種類の変数があります。
ほ	
ポイント・クエリ (point query)	結果を特定の値 1 つに制限する クエリ (query) 。通常は "where column_value = search_argument" のフォーマットを使用します。
ホーム・ページ (home page)	ローの元の位置。 ローの転送 (row forwarding) によってローが移動された場合、 ホーム・ページ (home page) にはローの 転送先アドレス (forwarding address) が格納されます。
ま	
マイナー・カラム (minor column)	複合インデックスのうち、先行カラムでないもの。(A、B、C) のインデックスでは、B と C がマイナー・カラムです。
マジック値 (magic values)	実際の値が不明な場合に、代用したり、デフォルトとして使用したりする値。たとえば、オプティマイザによってクエリが最適化される際に、パラメータの値の選択性が特定できなかった場合には、マジック値が代わりに使用されます。
マスタ・ディテール関係 (master-detail relationship)	1 セットのデータが、他のデータ・セットに論理的に依存するデータ・セット間の関係。たとえば、pubs2 データベースでは、sales テーブルと salesdetail テーブルの間にマスタ・ディテール関係があります。「 詳細 (detail) 」と「 マスタ・テーブル (master table) 」参照。
マスタ・テーブル (master table)	別のテーブルのデータが論理的に従属しているデータが含まれたテーブル。たとえば、pubs2 データベースでは、sales テーブルがマスタ・テーブルです。salesdetail テーブルには、sales 内のマスタ・データに従属するディテール・データがあります。ディテール・テーブルには、通常はマスタ・テーブルのプライマリ・キーにジョインするための外部キーがあります。

マッチング・インデックス・スキャン (matching index scan)	クエリのカラム・セットに where 句 (探索引数) があり、インデックス内のキーの プレフィクス・サブセット (prefix subset) が形成されるときに、ノンクラスタード・インデックスを使ったスキャン。最初のマッチング・キーの検索にインデックスを使用し、次に指定したインデックス・キー・カラムについてさらにマッチング・キーがないか前方スキャンをします。値が一致しない最初のローでスキャンは終了します。マッチング・インデックス・スキャンはきわめて高速で行われ、効率的です。「 非マッチング・インデックス・スキャン (nonmatching index scan) 」参照。
マテリアライズされた計算カラム (materialized computed column)	結果が事前評価され、指定したテーブルまたはインデックス・ページに格納される計算カラム。
幻ロー (phantom rows)	1 つ目のトランザクションが探索条件を満たすロー・セットを読み込んだ後、2 つ目のトランザクションがそのデータを、 insert 、 delete 、 update などを使って修正すると発生します。最初のトランザクションが同じ探索条件で読み込みを繰り返すと、別のロー・セットを得ることになります。幻ともいいます。
マルチデータベース・トランザクション (multi-database transaction)	複数のデータベースのデータに対するアクセス、使用、変更が行われるトランザクション。
マルチバイト文字セット (multibyte character set)	複数のバイトを使ってコード化した文字が含まれる 文字セット (character set) 。日本語言語環境のマルチバイトで表した複数の文字タイプを組み込んだ文字セットには EUC JIS と Shift-JIS などがあります。
マルチプロセス (multiprocessing)	メモリを共有したり、互いのメッセージのやりとりで同期方式を使う複数のプロセスはマルチプロセッシングを行います。
み	
密度 (density)	1 つのインデックス内のすべてのローが、同じ値を持つ平均比率。密度が 1 の場合、すべてのデータ値は同じであり、どのデータ値もユニークな場合、1/N になります。
密頻度カウント (dense frequency count)	ドメイン内で値が連続する頻度のカウント。たとえば、整数ドメインで 1、2、3 の値を設定すると高密度になります。「 疎頻度カウント (sparse frequency count) 」参照。
ミラー (mirror)	「 ディスク・ミラーリング (disk mirroring) 」参照。
ミラー・デバイス (mirror device)	複製 Adaptive Server データベース・デバイス。プライマリ・デバイスへの書き込みは、すべて 2 番目の物理デバイスにコピー (ミラーリング) されます。書き込みは、シリアル (連続) またはパラレル (同時) のいずれかにできます。一方のデバイスに障害が起きても、もう一方のデバイスにすべてのトランザクションの最新コピーが保管されています。

む	
結び付き (affinity)	特定の Adaptive Server タスクが特定のエンジン上だけで実行されるプロセス (タスクとの結び付き)、特定のエンジンが特定のタスク用のネットワーク I/O を処理するプロセス (ネットワーク I/O との結び付き)、または特定のエンジンが特定の CPU 上だけで実行されるプロセス (エンジンとの結び付き)。
め	
メイン・フォーム (main form)	アプリケーションが表示する最初のウィンドウ。アプレットやアプリケーションのターゲットを作成すると、メイン・フォームが作成されます。
メタデータ (metadata)	データに関するデータ。メタデータはコンポーネント統合サービスによってローカル・プロキシ・テーブルに保存されます。コンポーネント統合サービスで保存されたメタデータは、リモート・テーブルに関する情報のスキーマを表します。
メタデータ記述 (metadata description)	メモリ・データ構造。実行中や実行と実行の間のキャッシュ時のインデックス、オブジェクト、データベースのステータスを表します。
メタデータ・キャッシュ (metadata cache)	インデックス、オブジェクト、データベースに関する情報を追跡するために使われるメモリの予約領域。メタデータ・キャッシュのサイズは、インデックス、オブジェクト、データベースで使われる メタデータ記述 (metadata description) の数をもとに設定できます。
メッセージ番号 (message number)	エラー・メッセージをユニークに識別するための番号。
メッセージング (messaging)	イベントまたはバイト列を、通信経路を介して送信元から送信先まで送信する処理。
メディア障害 (media failure)	メディア障害はメディア (通常はハードディスク) 上の情報が使えなくなると発生します。
も	
文字式 (character expression)	文字型の値を 1 つ返す式。リテラル、連結演算子、関数、カラム識別子を組み合わせることができます。
文字セット (character set)	各文字をユニークに定義するコード化スキームを持つ特定の (通常、標準化された) 文字の集まり。ASCII と ISO 8859-1 (Latin 1) は、よく使用される文字セットです。
文字セット変換 (character set conversion)	Adaptive Server に対する文字セットのコード化スキームを途中で変換すること。変更は、Adaptive Server と、通信相手のクライアントが異なる文字セットを使っている場合に行われます。たとえば、Adaptive Server が ISO 8859-1 を使用し、クライアントが CodePage 850 を使用する場合、文字セット変換をオンにして、サーバとクライアントが、受け渡しされるデータを同じように解釈するようにします。

モジュール (modulo)	パーセント (%) 記号で表した 算術演算子 (arithmetic operator) 。2つの整数で除算したときの余りの整数値を提供します。次に例を示します。21 % 9 = 3 となるのは、21 割る 9 は、商が 2 で余りが 3 になることを表します。
文字列関数 (string function)	文字列やバイナリ・データに使用する関数。substring および charindex は、Transact-SQL 文字列関数です。
や	
役割 (roles)	Adaptive Server が認識するタイトル。Adaptive Server でシステム管理やセキュリティ関連のタスクを実行する個々の責任範囲をユーザに指定します。 システム管理者 (System Administrator) 、 システム・セキュリティ担当者 (System Security Officer) 、 オペレータ (Operator) の役割は、別々のサーバ・ログイン・アカウントに付与することができます。これらのシステム定義役割のほか、システム・セキュリティ担当者は、「財務分析者」、「給与管理者」など、ユーザ定義役割を作成することができます。
ゆ	
ユーザ ID (user ID)	特定のデータベースで、ユーザを認識するための ID 番号。「 サーバ・ユーザ ID (server user ID) 」参照。
ユーザ定義キャッシュ (user-defined cache)	「 バッファ・キャッシュ (buffer cache) 」参照。
ユーザ定義データ型 (user-defined datatype)	ユーザが作成する、 カラム (column) 内に含むデータの型 (データ型 (datatype)) の定義。これらの データ型 (datatype) は既存のシステム・データ型をもとに定義します。ルールとデフォルトはシステム・データ型にはバインドできませんが、ユーザ定義データ型にはバインドできます。
ユーザ定義トランザクション (user-defined transaction)	「 トランザクション (transaction) 」参照。
ユーザ・データベース (user database)	ユーザが作成するデータベース。「 システム・データベース (system database) 」参照。
ユーザ・テーブル (user table)	ユーザ・データを格納するデータベース・ テーブル (table) 。
ユニーク・インデックス (unique indexes)	指定した カラム (column) 内の2つの ロー (row) で同じ値を持つことを許さないインデックス。 Adaptive Server は、 インデックス (index) の作成時 (すでにデータが存在する場合) またはデータが追加されるたびに、重複値がないかどうかをチェックします。
よ	
予測ロー・サイズ (expected row size)	ユーザ定義のデータ・ロー予測サイズの平均。テーブルの予測ロー・サイズを指定することによって、データ・ページ上に、挿入された後に長さが増加するローのための領域が確保されます。これにより ローの転送 (row forwarding) が回避されます。

予備セット (look-ahead set)	非同期プリフェッチでフェッチされる特定の操作のためのページ・セット。非同期プリフェッチを使う Adaptive Server の各操作では、既知のまたは近い将来、操作で必要になるであろうと予測されるページ・セットに基づいて、予備セットが構築されます。
読み込みアクセス (read access)	オブジェクトを読み込むパーミッション (たとえばテーブルからローを選択する)。「 パーミッション (permission) 」参照。
予約語 (reserved word)	「 キーワード (keyword) 」参照。
予約ページ・ギャップ (reserve page gap)	バルク・コピー、 create index 、 select into によって実行される大規模な割り付け中に使用する割合。予約ページ・ギャップの値を設定することにより、ローの転送やインデックス・ページの分割のためのページ割り付けを、その基となるデータの場所の近くで行うことができます。
ら	
ライトウェイト・プロシージャ (lightweight procedure)	Adaptive Server の内部で作成して呼び出すことができるプロシージャ。ライトウェイト・プロシージャにはシステム・テーブル・エントリはありません。プロシージャ・キャッシュに保存されたプランとそれを識別するメモリ・データ構造で構成されます。
ライトウェイト・プロセス (lightweight process)	「 ワーカー・プロセス (worker process) 」参照。
ライブラリ (libraries)	「 アプリケーション・プログラム・インタフェース (application program interface: API) 」参照。
ライブロック (livelock)	重複した共有ロック (shared lock) により干渉が続いたために何度も拒否された 排他ロック (exclusive lock) に対する要求。 Adaptive Server では、3 回の拒否後に状況を検出し、更新トランザクションにデマンド・ロックを与え、デマンド・ロック後は後続の共有ロック要求をキューイングします。
ラウンドロビン分割 (round-robin partitioning)	負荷分散に最適な分割方式。データ値をどの位置で分けるかは考慮せず、ラウンドロビン方式でデータ・セットを分散します。この分割方式に関連付けられたセマンティックはありません。つまり、特定のローが特定のパーティションに格納される保証はありません。
ラストチャンス・スレッシュホルド (last-chance threshold)	Adaptive Server のデフォルトのスレッシュホルド。トランザクション・ログの容量がいっぱいになったらユーザ・プロセスを中断または強制終了します。このスレッシュホルドでは、ログ自体のレコードを割り付け解除ができるだけのスペースしか残りません。ラストチャンス・スレッシュホルドでは、 sp_thresholdaction というプロシージャが呼び出されます。このプロシージャは Sybase からは提供していません。システム管理者が作成する必要があります。
ラッチ (latch)	ページの物理的な整合性を保つために非常に短時間で行われる、ライトウェイトな非トランザクション型の同期メカニズム。

リ

リーフ・レベル (leaf level)	すべてのキーの値が順番に表示されるインデックスのレベルです。Adaptive Server クラスタード・インデックスの場合、リーフ・レベルとデータ・レベルは同じです。ノンクラスタード・インデックスの場合、データ・レベルの上の最終インデックス・レベルがリーフ・レベルです。これはすべてのデータ・ローのキー値は、ここにソート順に表示されるからです。
リカバリ (recovery)	データベースとログのダンプ (dump) から 1 つ以上のデータベースを再構築するプロセス。「自動リカバリ (automatic recovery)」参照。
リスト分割 (list partitioning)	分割キーの値に基づく、テーブル (table) またはインデックス (index) の分割。ユーザが指定した値と 1 つまたは複数の分割キーが比較され、各ロー (row) が属するパーティション (partition) が決定されます。
リソース予約 (resource reservation)	(クラスタの概念) 特定の論理クラスタ用に 1 つのインスタンスを確保し、その論理クラスタにルートされたクライアントのみにそのインスタンスへの接続を許可すること。リソース予約を行うには、システム論理クラスだけでなく、論理クラスタにもオープン・プロパティを割り当てる必要があります。
リターン・ステータス (return status)	プロシージャが正常に実行されたことや、失敗した理由を示す値。
リモート・プロシージャ・コール (remote procedure call: RPC)	ユーザがログインしているサーバ以外の Adaptive Server で実行するストアード・プロシージャ (stored procedure)。
リモート・ログイン (remote login)	リモート・サーバへのログイン。
領域の再利用 (space reclamation)	ローを縮小する削除または更新の後で、データオンリー・ロック・テーブルから領域を回復するプロセス。領域の再利用は、ページ上のローを挿入したり更新したりするタスクや、ハウスキーピング、reorg コマンドによって実行されます。
リラックス持続性データベース (relaxed-durability database)	ディスク上に常駐し、持続性レベルが no_recovery または at_shutdown であるデータベース。リラックス持続性データベースは、ディスク上に常駐する full 持続性を持つ従来のデータベースがサポートするすべての機能 (キャッシュのバインドなど) をサポートします。リラックス持続性データベースでは、その持続性要件が低いことを利用して、トランザクションの負荷の実行時に従来のディスク常駐のデータベースよりも大幅に改善されたパフォーマンスが実現できます。
リレーショナル・データベース (relational database)	特定目的に対応するために構成された、関連データ・テーブルおよびその他のデータベース・オブジェクト (database object) の集まり。「データ・ウェアハウス (data warehouse)」参照。

リレーショナル・データベース・マネジメント・システム (Relational Database Management System: RDBMS)	SQL の使用を標準とし、2次元テーブルに対してデータを保存および検索するシステム。
る	
ルール (rule)	特定の カラム (column) や特定の ユーザ定義データ型 (user-defined datatype) のカラムに、どのようなデータを入力できるかを制御する仕様。
れ	
レガシー・データ (legacy data)	古くなった、おそらく時代遅れのデータ・ソースのデータ。または現在の標準でサポートされていないデータ・ソースのデータ。
レコード (record)	「 ロー (row) 」参照。
連結 (concatenation)	式を結合して長い式を作成すること。式では、バイナリ列か文字列、またはカラム名を任意に組み合わせることができます。
連鎖トランザクション・モード (chained transaction mode)	Adaptive Server が、次のデータ検索文やデータ変更文に対して新しいトランザクションを自動的に始めるかどうかを決定します。連鎖トランザクション・モードがトランザクションの外側でオンに設定されていると、次のデータ検索文やデータ変更文は新しいトランザクションを開始します。このモードは ANSI 準拠です。このモードでは、SQL データ検索文やデータ変更文は、トランザクションの内側で発生します。連鎖トランザクション・モードは、既存の Transact-SQL プログラムとの互換性はありません。連鎖トランザクション・モードは、デフォルトではオフに設定されています。ANSI SQL (Embedded SQL プリコンパイラなど) を必要とするアプリケーションでは、各セッションの開始時に自動的に chained オプションをオンにするように設定します。
連邦情報処理規格 (FIPS) フラガ (Federal Information Processing Standards flagger)	set コマンドで起動する Adaptive Server オプション。FIPS フラガが有効なときに初級レベル SQL92 の Transact-SQL 拡張機能を使うと、 Adaptive Server では警告メッセージが表示されます。FIPS では SQL89 を基本標準として認識します。
る	
ロー (row)	特定の エンティティ (entity) を記述する、関連 カラム (column) の集まり。 レコード (record) ともいいます。
ロー ID (row ID: RID)	データ・ ロー (row) 用のユニークな内部識別子。ロー ID (RID) は、データ・ページ番号とそのページのロー番号を組み合わせたものです。
ローカライゼーション (localization)	国際化された製品をある特定の言語または地域の要件に合うように適応させること。これには、その国の言語に翻訳されたシステム・メッセージやユーザ・インタフェース、その国で使用している日付、時間、通貨の正しいフォーマットを提供することも含まれます。

ローカライゼーション・ファイル (localization files)	翻訳されたエラー・メッセージ、文字セットやソート順の定義、 Adaptive Server の文字セットを特定の端末に適した文字セットに変換するためのユーティリティを含むファイル。
ローカル・インデックス (local index)	テーブルの インデックス (index) が、テーブルのデータと同じ方法で分割される場合を指します。
ローカル・サーバ (local server)	クエリ (query) の発生元のサーバまたは ノード (node) 。
ローカル・システム・テンポラリー・データベース (local system temporary database)	(クラスタの概念) テンポラリー・テーブルとテンポラリー・ワーク・テーブル用の領域。クラスタ内の各インスタンスには、そのインスタンスのみがアクセスできるローカル・システム・テンポラリー・データベースが用意されています。
ローカル統計 (local statistics)	分割されたテーブルの特定のパーティションのデータ値に適用される統計。
ローカル変数 (local variable)	declare 文で定義するユーザ定義の 変数 (variable) 。「 グローバル変数 (global variable) 」参照。
ロー集合関数 (row aggregate function)	select 文 (statement) の compute で使用し、計算データ用に新しい ロー (row) を生成する関数 (sum、avg、min、max、count)。
ロード (load)	ダンプ (dump) 時に作成されたバックアップに保存されたデータをリストアすること。
ローの転送 (row forwarding)	他のページへのローの移動。ローが更新されてページに適合しなくなった場合に、ローの転送が行われます。 転送されたロー (forwarded row) へのポインタは、元のローの位置で保持されます。そのローのロー ID は、変更されません。
ロールバック・トランザクション (rollback transaction)	commit transaction を受け取る前に ユーザ定義トランザクション (user-defined transaction) で使用された Transact-SQL 文 (statement)。トランザクションを取り消し、データベースに対して加えられたすべての変更を元に戻します。
ロールバック・レコード (rollback record)	Adaptive Server が生成するログ・レコードであり、トランザクションによる事前の変更を取り消すための情報が書き込まれています。ロールバック・レコードには、取り消しを行うためのログ・レコードのアドレスが含まれています。ロールバック・レコードは、変更を取り消すためだけのログ・レコードです。ロールバック・レコード自身が取り消されることはありません。
ログイン (login)	ユーザが Adaptive Server へのログインに使用する名前。 Adaptive Server のシステム・テーブル syslogins にそのユーザ用のエントリが存在すれば、ログインが有効になります。
ログインのリダイレクト (login redirection)	(クラスタの概念) インスタンスが、着信クライアント接続をクラスタ内の異なるインスタンスにダイレクトするメカニズム。ログインのリダイレクトは、インバウンド接続を論理クラスタ内のインスタンスにルーティングして負荷のバランスをとるために使用されます。

ロケーション・トランスパレンシ (location transparency)	ローカルに存在しているようにリモート・データを表示させる、コンポーネント統合サービス固有の機能。アクセスするとき、ユーザはそのデータがどこにあるかを意識せずに済みます。
ロッキング (locking)	マルチユーザ環境においてリソースのアクセスを制限するプロセス。セキュリティを保持し、同時アクセス問題を防ぎます。Adaptive Server によるテーブル、ページ、またはローへのロックは自動的に行われます。
ロック (lock)	マルチユーザ環境におけるデータとトランザクション結果の整合性を保護する同時実行制御メカニズム。Adaptive Server は、2 人のユーザが同じデータを同時に変更しようとする、ページ・ロックまたはテーブル・ロックを適用します。また、プロセス変更中のデータから読み込んだデータも選択できません。
ロック・プロモーション (lock promotion)	スキャンによって取得されたロー (row) またはページ・ロック (page lock) を、テーブル・ロックにまで拡大すること。スキャンでロック・プロモーション・スレッシュホールドのユーザ設定が可能な範囲を超えると、ロック・プロモーションが発生します。ロックはロー・ロックからテーブル・ロックへ、またはページ・ロックからテーブル・ロックへと拡大します。
ロック・プロモーション・スレッシュホールド (lock promotion threshold)	Adaptive Server がテーブル・ロックを行う前に、1 つのテーブルで許可するページ・ロックの数。テーブル・ロックが正常に行われると、Adaptive Server はページ・ロックを解除します。
ロック・レベル (locking level)	「独立性レベル (isolation level)」参照。
論理 I/O コスト (logical I/O cost)	オプティマイザが実行する論理読み込みの回数の見積もり。
論理演算子 (logical operator)	演算子 (operators) and、or、および not。以上の 3 つは、すべて where 句で使用できます。演算子 and では複数の条件が結び付けられ、すべての条件が満たされたときに結果が返されます。or では複数の条件が結び付けられ、どれかひとつでも条件が満たされると結果が返されます。最適化における論理演算子は、クエリ処理の特定のアルゴリズムを指定しないオペレーション (ジョイン、スキャン、ソート) です。
論理キー (logical key)	データベース設計上のプライマリ・キー、外部キー、または共通キーの定義。データベースのテーブル間の関係を定義します。論理キーは、テーブル上の物理キー (physical key) (インデックスの作成に使用されるキー) と必ずしも同じである必要はありません。
論理クラスタ (logical cluster)	(クラスタの概念) 複数アプリケーション・サービスが確立できるように物理クラスタを抽象化する方法。論理クラスタは、アプリケーション固有またはユーザ固有のサービス・レベル・アグリーメント、リソース割り当て、およびフェールオーバー・ルールを使用することで、クラスタ内の負荷の微調整管理をサポートしています。アプリケーションは、論理クラスタに直接接続します。
論理式 (logical expression)	評価することによって、TRUE (1)、FALSE (0)、UNKNOWN (NULL) を判断する式。論理式は if 条件や while 条件などのフロー文の制御に頻繁に使用されます。

論理設計 (logical design)	リレーショナル・データベースのテーブル、関係、キーを定義すること。「物理設計 (physical design)」参照。
論理プロパティ (logical property)	一連のテーブル (同等クラス) と関連付けられた一連のサブプランに共通するプロパティ。ロー・カウントはその一例です。これは、テーブルのセットのジョイン方法に関係なく、同じ述部の適用後は同じローが存在するためです。
論理分割 (logical partitioning)	ローを2つ以上のパーティションに分割すること。Adaptive Server では、セマンティック分割 (ライセンスが必要) またはラウンドロビン分割 (ライセンスは不要) のいずれかを使用することを指します。
論理ページ・サイズ (logical page size)	新しいマスタ・デバイスを構築するときに指定する。すべてのデータベース、および、各データベースにあるすべてのオブジェクトには、同じサイズの論理ページが使用されます。Adaptive Server の論理ページのサイズ (2、4、8、または 16K) によって、サーバの領域の割り付けが決まります。
論理読み込み (logical read)	クエリの結果を返すのにメモリ内にあるデータやインデックス・ページをアクセスする処理。「物理読み込み (physical read)」参照。
わ	
ワーカー・プロセス (worker process)	並列クエリ処理環境で Adaptive Server タスクから生成される Adaptive Server サブタスク。他のワーカー・プロセスとともにクエリ (query) の一部を同時に実行し、応答時間を短縮します。ワーカー・プロセスは、プロセスに必要なオペレーティング・システム・リソースを少ししか使用しません。起動後にはオペレーティング・システム・リソースを必要としません。相互にメモリ・スペースを共有します。スレッド (thread)、またはライトウェイト・プロセスともいいます。
ワーク・テーブル (worktable)	「動的インデックス (dynamic index)」参照。
ワイルドカード文字 (wildcard character)	Transact-SQL like キーワードとともに使用する特殊文字。パターン一致で1つまたは任意の文字を表すことができます。

