第2章

データベースの構成

ここでは、Oracle、SQL Serverのデータベースレベルの機能について記述します。

システムデータベースとユーザーデータベース

Oracleはインスタンスに対して1つのデータベースという構成で、インスタンスの構成情報は初期化パラメータファイル（INIT<SID>.ORA）で設定され、データベースの構成情報は制御ファイルおよびデータディクショナリにあります。

SQL Serverは、1つのインスタンスに対して複数のデータベースを構成できます。インスタンスの構成情報はシステムデータベースにあり、データベースの構成情報はそれぞれのユーザーデータベースのシステムテーブルにあります。各システムデータベースの役割は、以下のとおりです。

- master
  SQL Serverのインスタンスレベルの構成情報、ログインアカウント、masterデータベース以外のデータベースの物理的情報を格納するデータベースです。
- tempdb
  ユーザーがデータを並べ替えるときなどに使用する作業用のデータベースです。すべてのデータは、SQL Serverの起動時に自動的に初期化されます。
- model
  データベースを作成するときにコピーされるデータベースです。テンプレートデータベースとも呼ばれ、ユーザーがカスタマイズすることも可能です。このデータベースがあるおかげで、データベースを高速に作成できるようになっています。
- msdb
  ジョブのスケジューリング、警告、オペレータ情報を格納するデータベースです。SQL Serverエージェントが使用します。

このほかに、SQL Serverをインストールすると、pubsとNorthwindというサンプルデータベースが作成されます。

図2-1は、1つのSQL Serverインスタンスに存在するデータベースを示しています。UserDB1とUserDB2はインストール後に作成したユーザーのデータベースです。

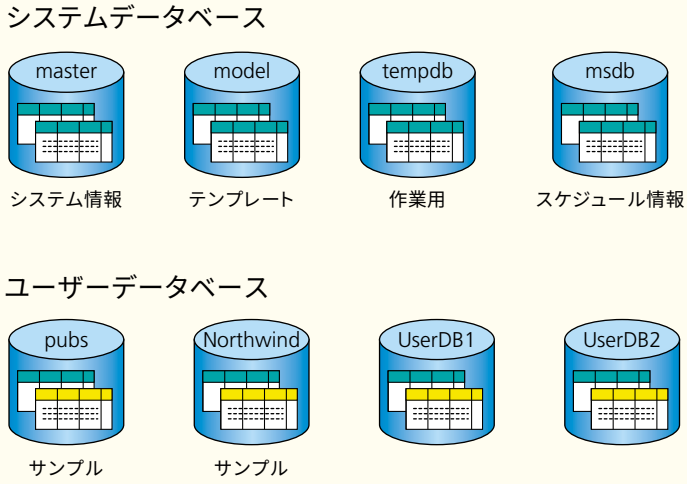


図2-1
システムデータベースと
ユーザーデータベース

データベースファイル

Oracleの基本的なデータベース構成ファイルは、制御ファイル、データファイル、REDOログファイルですが、SQL Serverではデータファイルとトランザクションログファイルです。SQL Serverで制御ファイルに当たる情報は、システムデータベースおよびシステムテーブルに格納されています。

図2-2は、OracleとSQL Serverのデータベースファイルの違いを示しています。

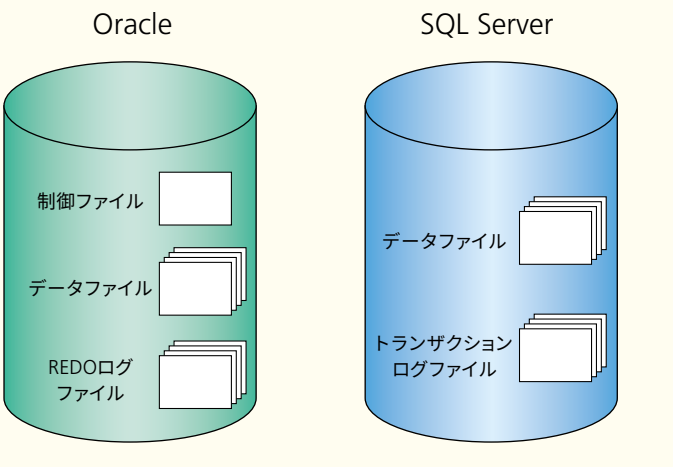


図2-2
データベース ファイル

SQL Serverのデータベース作成時には、各ファイルに論理ファイル名、物理パス、初期サイズ、拡張方法を指定します。

表：データベース作成時の設定値例

項目	値
データベースの名前	test
データベースファイルの論理ファイル名	test_data
データベースファイルの物理パス	E:¥MSSQL¥test_data.mdf
初期サイズ	1Gバイト
ファイルの自動拡張	ON
ファイルの自動拡張単位	ファイルの大きさの10%
ファイルの自動拡張の最大サイズ	無制限（Eドライブのすべての空き領域を使用）

図2-3は、SQL Server Enterprise Managerを使用してデータベースを作成しているときの画面です。

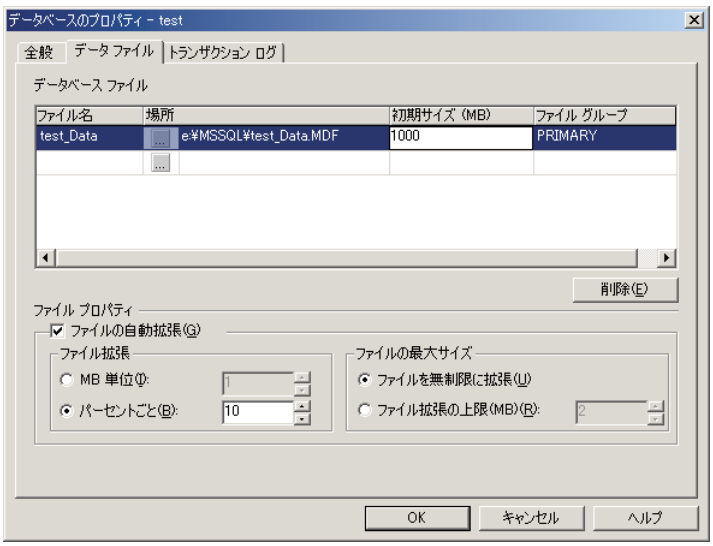


図2-3
データベースの作成
（SQL Server Enterprise Manager）

データファイルを細かく分類すると下記の2種類があります。

- **プライマリデータファイル**
  プライマリデータファイルはデータベースの開始点であり、データベース内のほかのファイルを指し示します。1つのデータベースに1つのプライマリデータファイルが必要です。推奨されているプライマリデータファイルの拡張子は.mdfです。
- **セカンダリデータファイル**
  セカンダリデータファイルは、プライマリデータファイル以外のすべてのデータファイルです。データベースは、セカンダリデータファイルがない場合と、複数のセカンダリデータファイルがある場合があります。推奨されているセカンダリデータファイルの拡張子は.ndfです。

ファイルグループ

Oracleのテーブルを割り当てる単位である表領域は、SQL Serverでのファイルをグループ化したファイルグループに相当します。ファイルグループは、表領域と同様にテーブルやインデックスなどを割り当てられる単位となります。データベース作成時に必ず作成されるファイルグループはプライマリファイルグループと呼ばれ、システムテーブルが格納されます。このプライマリファイルグループはOracleのシステム表領域とほぼ同等です。

図2-4は、Oracleの表領域とファイルの関係、およびSQL Serverのファイルグループとファイルの関係を示したものです。

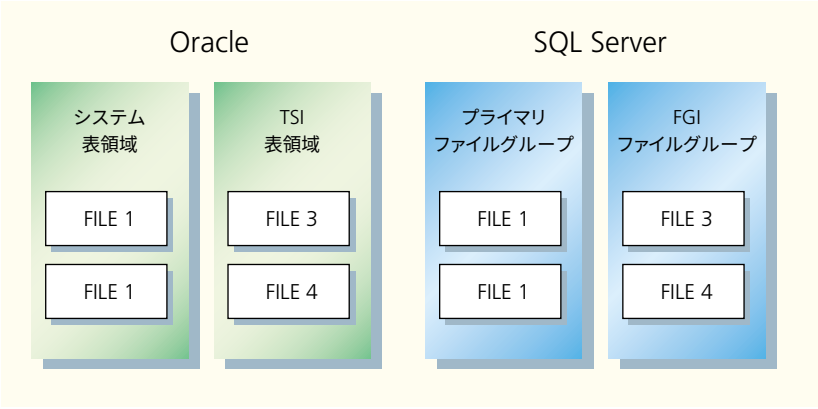


図2-4
ファイルグループ

SQL Serverのファイルグループに複数のファイルを割り当てた場合は、自動的に64Kバイトごとに分散して（プロポーショナルファイル方式）データを格納します。これらのファイルを物理的に違うハードディスクに分散させることにより、SQL Serverレベルでの擬似的なRAID 0（ストライピング）が可能となります。すべてのファイルが同じ空き領域の場合は、図2-5のように均等に書き込みを行います。異なる空き領域の場合には、大きさの割合に応じて書き込む順番を変えます。Oracleの場合は、パーティションを作成する必要があります。

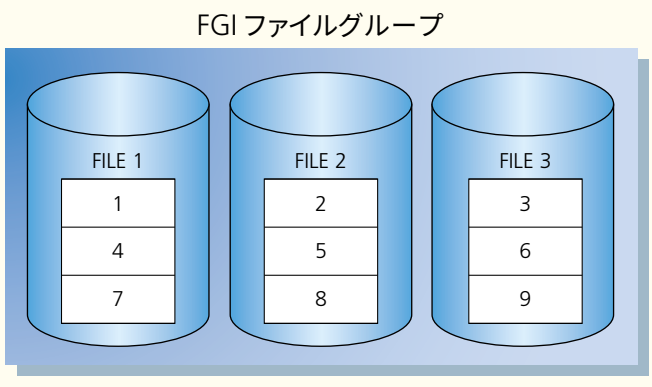


図2-5
ファイルグループに
複数ファイルを設定したときの
書き込みの順番

データベース内のいずれか１つのファイルグループがデフォルトファイルグループとなります。デフォルトファイルグループは、ユーザーがテーブルやインデックスなどを作成するときに明示的にファイルグループを指定しなかった場合に使用されるファイルグループです。特に指定がなければプライマリファイルグループがデフォルトファイルグループになりますが、別のファイルグループに変更することも可能です。また、プライマリファイルグループ以外は読み取り専用として設定することも可能です。

エクステントとページ

Oracleのデータファイルは、エクステントとブロックに分けて管理されています。ブロックはデータベース内では固定長（ブロックサイズは2K、4K、8K、16Kなどで、オペレーティングシステムブロックで構成されます）で、エクステントサイズはセグメント単位に設定可能です。

SQL Serverのデータファイルは、エクステントとページに分けて管理されています。ページが８個連続で集まってエクステントになります。サイズはWindowsオペレーティングシステムに最適に設計された固定の長さを持ち、ページは8Kバイトで、エクステントは８倍の64Kバイトです。固定長であるため、データベースファイル内の管理が簡略化され、高速にデータページにアクセスできます。

図2-6は、Oracleのブロックとエクステントおよび SQL Serverのエクステントとページの違いを図示しています。

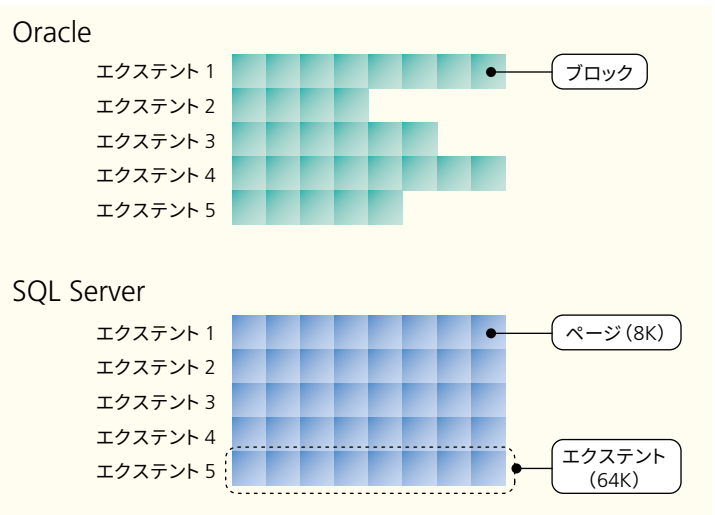
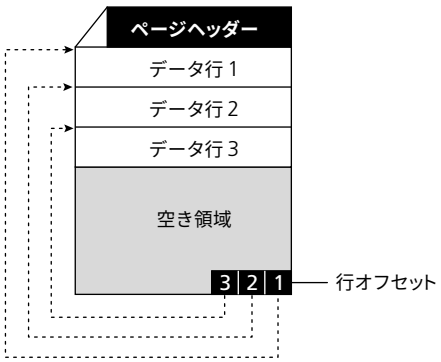


図2-6
エクステントとページ

ページ

SQL Serverのページの特徴は、以下のとおりです。

- メモリとハードディスクの最低入出力単位です。
- 1つのページには必ず1つのオブジェクトデータだけが入っています。
- テーブルの各1行は複数のページにまたがることはできません。ただし、OracleのCLOBおよびBLOBデータ型に当たるtext、ntext、imageデータ型は含みません。
- ファイルの先頭から各ページにページ番号が付いています。
- ページの先頭は、96バイト長の固定のヘッダーであり、このヘッダーには、そのページのタイプ、空き領域のサイズ、そのページを所有するオブジェクトのオブジェクトIDなどのシステム情報が格納されています。
- SQL Serverでは、各行のデータは上から埋められていき、ページの最後から各行のオフセットが格納されます（下図参照）。Oracleでは、ブロックヘッダーが可変長のため、各行のデータは下から埋められます。



- Oracleでは、セグメント（表セグメント、索引セグメントなど）ごとにブロックの用途が違いますが、SQL Serverはページ単位で用途が異なり、次の表に示しているようなページタイプがあります。

ページタイプ	内容
データ	text、ntext、imageの各データ型を除くすべてのテーブルのデータが入っているページ
インデックス	インデックスエントリデータが入っているページ
テキスト/イメージ	text、ntext、imageの各データ型のデータが入っているページ
グローバルアロケーションマップ、セカンダリアロケーションマップ	割り当てられたエクステントに関する情報が入っているページ

（続き）

ページタイプ	内容
ページ空き領域	各ページ上で利用可能な空き領域に関する情報が入っているページ
インデックスアロケーションマップ	各テーブルまたはインデックスによって使用されるエクステントに関する情報が入っているページ
一括変更マップ	最後のBACKUP LOGステートメント以降の、一括操作で変更されたエクステントに関する情報が入っているページ
差分変更マップ	最後のBACKUP DATABASEステートメント以降に変更されたエクステントに関する情報が入っているページ

エクステント

SQL Serverのエクステントは、テーブルとインデックスに領域を割り当てる基本単位です。領域の割り当てを効率的にするために、データ量が少ないテーブルにエクステント全体を割り当てることはありません。SQL Serverには2種類のエクステントがあります。

- **単一エクステント**
  単一のオブジェクトに所有され、所有しているオブジェクトだけがエクステント内の8ページすべてを使用できます。
- **混合エクステント**
  エクステント内の各ページは複数のオブジェクトが所有します。最大8つのオブジェクトに共有されます。

通常、新規のテーブルまたはインデックスには、混合エクステントからページが割り当てられます。そのテーブルまたはインデックスが8ページまで拡張した時点で、単一エクステントに切り替えられます。既存のテーブルに対してインデックスを作成する場合は、インデックスに8ページ分を生成できるだけの行があれば、インデックスへのすべての割り当ては単一エクステントです。

空き領域の管理

Oracleでは、空きリストを利用してデータベース内の空き領域を管理しています。

SQL Serverでは、エクステントとページの空き領域を管理するページがあ

ります。SQL Serverでエクステントの割り当てを記録するには、GAMとSGAMという2種類のページを使用します。

- **GAM（Global Allocation Map）**
  GAMページには、どのエクステントがすでに割り当てられているかが記録されます。1つのGAMで64,000エクステント、つまり約4Gバイトのデータが対象となります。GAMは、対象とする範囲内の各エクステントにつき1ビットを割り当てています。このビットの値が1の場合、そのエクステントは空いており、このビットの値が0ならば、そのエクステントは割り当て済みです。
- **SGAM（Shared Global Allocation Map）**
  SGAMページには、その時点でどのエクステントが混合エクステントとして使用されており1ページ以上の未使用ページを含んでいるかが記録されます。1つのSGAMで64,000エクステント、つまり約4Gバイトのデータが対象となります。SGAMは、対象とする範囲内の各エクステントにつき1ビットを割り当てています。このビットの値が1の場合、そのエクステントは混合エクステントとして使用されており、空きページを含んでいます。このビットの値が0ならば、そのエクステントは混合エクステントとして使用されていないか、全ページが使用されている混合エクステントです。

各エクステントは、現在の使用状況に基づいてGAMおよびSGAM内に次のようなビットパターンが設定されます。

エクステントの現在の使用状況	GAMのビット設定	SGAMのビット設定
空きで未使用	1	0
単一エクステント、または全ページを使用した混合エクステント	0	0
空きページがある混合エクステント	0	1

あるオブジェクトにエクステントを割り当てる必要がある場合は、GAMのビットの値が1でSGAMのビットの値が0であるエクステントを割り当てます。あるオブジェクトに混合エクステントのページを割り当てる必要がある場合は、GAMのビットの値が0でSGAMのビットの値が1であるエクステントのページを割り当てます。該当するものがない場合はファイルの拡張が発生します。

PFS（Page Free Space）ページには、各ページ、またはntext、text、image列が割り当て済みであるかどうかだけでなく、各ページの空き領域のサイ

ズも記録されます。1つのPFSページで約8,000 ページが対象となります。PFS には、ページごとに1つのビットマップがあり、そのビットマップには、そのページが空きか、1～50%が満たされているか、51～80%が満たされているか、81～95%が満たされているか、それとも96～100%が満たされているかが記録されています。

PFSページはデータファイル内でファイルヘッダーページの後ろにある最初のページであり、ページ番号は1です。その次にGAM（ページ2）があり、その後ろにSGAM（ページ3）があります（図2-7）。

最初のPFSページの後ろ、約8,000ページごとに1つのPFSページがあります。ページ2の最初のGAMの後ろには64,000エクステントごとに1つのGAMがあり、ページ3の最初のSGAMの後ろには64,000エクステントごとに1つのSGAMがあります。たとえば、512Mバイトのデータファイルの場合は、272Kバイトがページとエクステントの空き領域の管理に使用されることになります。

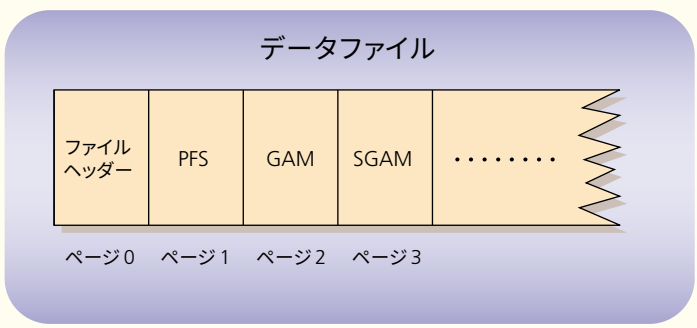


図2-7
空き領域を管理するページ

データベースの接続制限

Oracleのデータベースのオプションには、RESTRICTオプションや読み取り専用のオプションがあります。

SQL Serverでは、以下のオプションがあります。

- **マルチユーザー**
  デフォルトの設定で、許可されているすべてのユーザーが接続可能です。
- **RESTRICTED**
  db_owner、dbcreator、sysadminの各ロールのメンバだけがデータベースを使用できます。
- **シングルユーザー**
  データベースにアクセスできるのは一度に1人のユーザーだけです。

- **読み取り専用**
  すべてのユーザーがデータベースを読み取ることしかできません。

図2-8は、SQL Server Enterprise Managerで接続制限を変更している画面です。

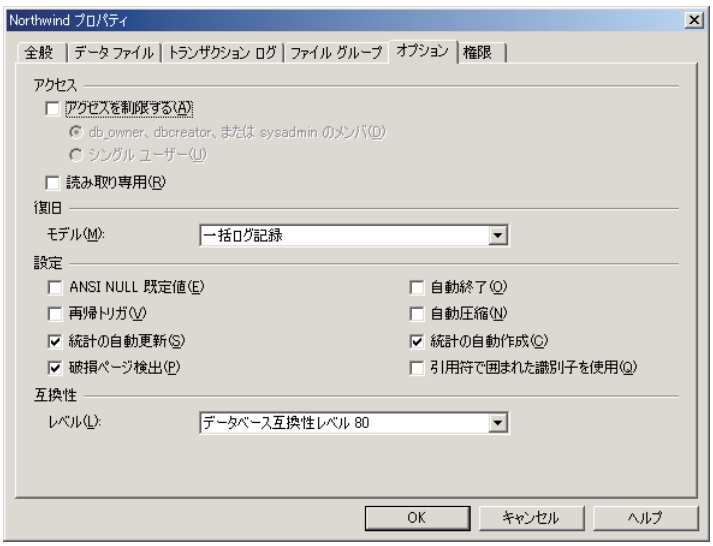


図2-8
データベースのオプション
(SQL Server Enterprise Manager)

トランザクション処理

SQL Serverがデータベースに変更を加える手順は、以下のとおりです(図2-9)。

① SQL Serverがデータの変更情報を受け取ると、該当するページをバッファキャッシュ内に読み込みます。そのとき使用されるバッファページは、空きバッファリスト内の先頭のページを使用します。空きバッファリストとは未使用のバッファページのアドレスで構成される、一重リンクのリストのことです。

② データキャッシュ内を変更するとそれに対応するログレコードがログキャッシュ内にログレコードとして記録されます。

③ ログレコードがトランザクションログに書き込まれます。通常これは、ログライタスレッドによって非同期にスケジューリングされます。ただし、コミットあるいはチェックポイントで強制的にトランザクションログに書き込む場合もあります。

④ データキャッシュ内の変更ページがデータファイルに書き出されます。この処理は、データベースにチェックポイントが実行されるか、空きバッファリストが少なくなり新しいページが必要になったときに行われます。なお、変更されたデータページをバッファキャッシュからディスクに書

き込むことを「ページをフラッシュする」と言います。キャッシュ内で修正され、まだディスクに書き込まれていないページのことを、「ダーティページ」と呼びます。

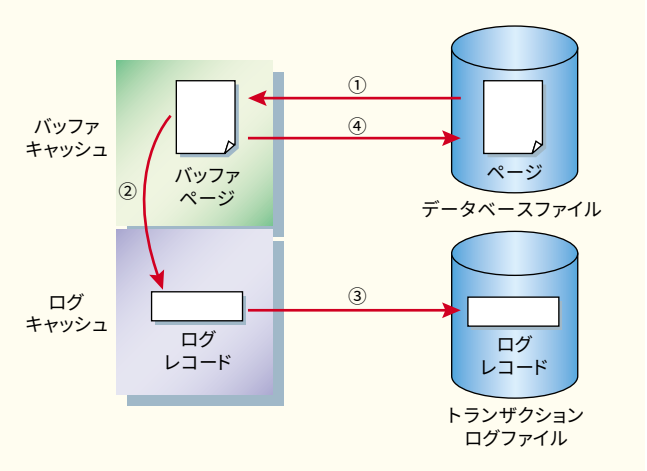


図2-9
トランザクション処理

チェックポイント

Oracleでチェックポイントを発生させる間隔を設定するパラメータには、以下のものがあります。

- DB_BLOCK_MAX_DIRTY_TARGET
- FAST_START_IO_TARGET
- LOG_CHECKPOINT_INTERVAL
- LOG_CHECKPOINT_TIMEOUT

SQL Serverでチェックポイントを発生させる間隔を設定するサーバーのオプションは、recovery interval（復旧間隔）です。recovery intervalは、インスタンス障害後の再起動時にデータベースの復旧に必要とする時間の最大値を、データベースごとに分単位に設定します。デフォルトは0です。0の場合、SQL Serverによって自動的に復旧間隔が設定されます。実際には、復旧時間が1分未満で、アクティブなデータベースのチェックポイントは約1分間隔になります。前回のチェックポイント後にデータベース内で行われたデータ変更の数が、復旧間隔内でロールフォワードできるとSQL Serverが予測すると、SQL Serverはデータベース内でチェックポイントを実行します。これらの予測は、SQL Serverがハードディスクの入出力速度などから自動的に計算します。さらに、チェックポイントは以下の場合にも発生します。

- **サーバーが正常に停止する場合**
  すべてのデータベースでチェックポイントが実行されます。
- **手動でチェックポイントを実行する場合**
  CHECKPOINT ステートメントを手動で実行します。
- **ALTER DATABASE ステートメントで**
  データベースオプションを変更する場合
  オプションが変更されるデータベースの中でチェックポイントが実行されます。

トランザクションログファイル

Oracle は、各トランザクションの復旧にロールバックセグメントを使用し、インスタンス障害やデータベースの復元時には REDO ログファイルを使用します。

SQL Server のトランザクションログの機能は、以下の 3 つです。

- **個々のトランザクションの復旧**
  アプリケーションが ROLLBACK ステートメントを実行するか、SQL Server がクライアントとの通信停止などのエラーを検出した場合は、未完のトランザクションによって加えられた修正をロールバックするためにログレコードが使用されます。
- **インスタンス障害の復旧**
  SQL Server を実行しているサーバーに障害が起きると、一部の修正がバッファキャッシュからデータファイルに書き込まれない状態で残る場合があります。つまり、未完のトランザクションからの修正がデータファイル内に保存されている可能性があります。SQL Server は、障害後に起動されると各データベースの復旧を行います。ログに記録されてデータファイルに書き込まれなかった可能性があるすべての修正がロールフォワードされます。その後、トランザクションログに入っている未完のトランザクションは、データベースの整合性を維持するために、すべてロールバックされます。
- **復元したデータベースの復旧**
  サーバーのハードドライブが損傷するなどの何らかの障害が発生した場合、障害が発生した時点までデータベースを復元できます。データベースを復元するには、まず最新のフルデータベースバックアップまたは差分データベースバックアップを復元し、次に一連のトランザクションログバックアップを障害が発生した時点まで復元します。各トランザクションログバックアップの復元では、ログに記録されている変更が SQL Server によって再適用され、すべてのトランザクションがロールフォワ

ードされます。最後のトランザクションログバックアップまで復元されると、今度はログ情報を使用して、障害の時点では完了していなかったすべてのトランザクションをロールバックします。

トランザクションログの切り捨て

Oracle の REDO ログファイルは複数のファイルで構成され、循環して使用されます。ARCHIVELOG モードを設定すると、上書きされる前に自動的にコピーが行われます。

SQL Server のトランザクションログも複数のファイルを使用しますが、データベースファイルと同様にラウンドロビン方式で使用されるため、論理的にはシーケンシャルにログレコードが書き出されます。トランザクションログの切り捨てを行わない限り、ファイルは増大していきます。トランザクションログの切り捨てを行うと非アクティブ部分は再利用されます（図 2-10）。非アクティブ部分とは、チェックポイントの発生時に完全にデータベースにコミットされたログレコードの集まりです。この循環方式は、Oracle のロールバックセグメントに似ています。

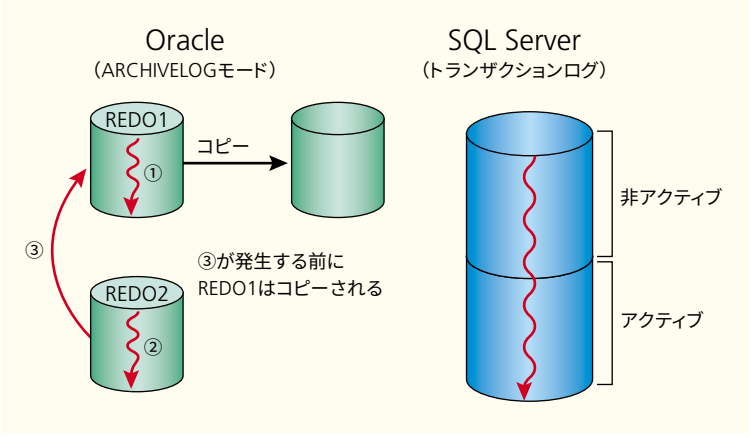


図 2-10
トランザクションログの
切り捨て

ログレコードには、ログシーケンシャル番号（LSN）が付きます。チェックポイントが発生するとログの内容とデータベースに書き込まれた内容が確認され、最小復旧 LSN（MinLSN）が記録されます。MinLSN より小さい LSN を持つログはデータベースに書き込まれていることが保証されており、非アクティブ状態のログレコードです。また、トランザクションログファイルは、論理的に仮想ログファイルと呼ばれるセクションに分割されます。

図 2-11 の①はデータベースを作成したあとに、ある程度の更新を行ったときのトランザクションログの状態です。ここでは、トランザクション論理ログの最後のレコードまでログが書き込まれています。このまま更新を続けて論理ログの最後が物理ログの最後に達すると、トランザクションログフ

ァイルは拡張されます。①の状態でトランザクションの切り捨てを行うと②の状態になります。②の状態からさらにデータベースを更新すると、論理ログの最後は物理ログの最後に達しますが、仮想ログ１が切り捨て済みのため再利用されます。再利用された状態が③になります。③の状態でトランザクションログの切り捨てを行うと仮想ログ３と仮想ログ４が再利用可能になります。

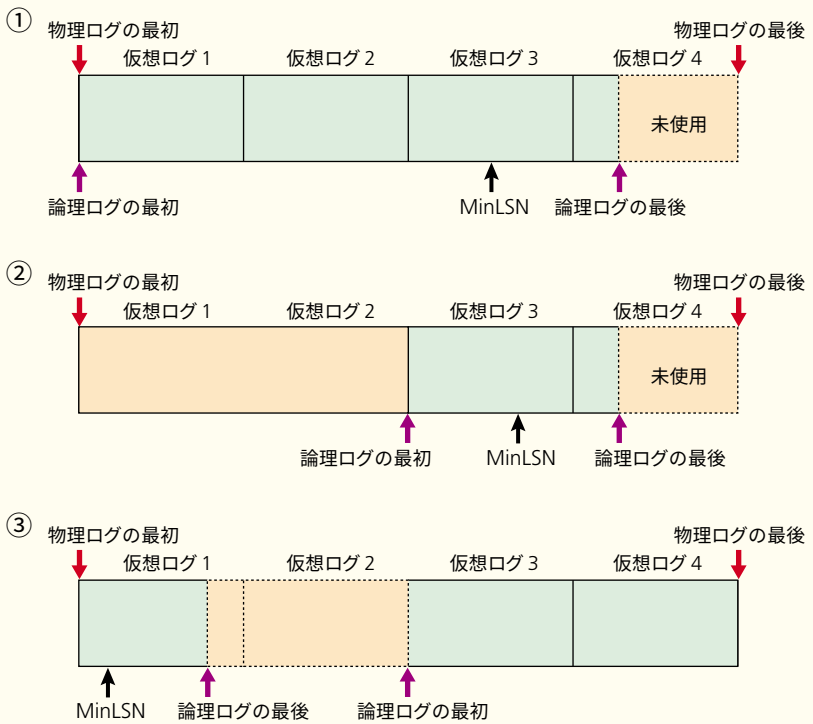


図 2-11
トランザクションログの
切り捨て（詳細）

トランザクションログの切り捨ては、以下のケースで発生します。

- **データベースの復旧モデルが単純（シンプル）の場合**
  チェックポイント時に自動的にトランザクションを切り捨てます。通常、チェックポイントは頻繁に行われるため、トランザクションログの容量はほとんど増えません。
- **データベースの復旧モデルが完全（フル）あるいは一括ログの場合**
  トランザクションログのバックアップのときにトランザクションを切り捨てます。トランザクションログファイルの拡張を防ぐためにトランザクションログのバックアップをスケジューリングする必要があります。スケジューリングの間隔とデータベースの変更状況を考えてトランザクションログファイルの容量を決めるようにしてください。