

Red Brick Warehouse Backup and Recovery

2000 年 8 月 1 日

Informix Software, Inc.

この文書はインフォミックス及びインフォミックスパートナー様限となっております。
取扱いには十分ご注意ください。

目次

目次	2
はじめに	3
RED BRICK のバックアップ/リストア	4
RED BRICK の概要 4	
バックアップ/リストア	6
バージョンが有効な場合のバックアップ/リストア	10
バージョンによるオンライン バックアップ	11
整合性のない状態への復旧	11
概要	12

はじめに

このマニュアルは、バックアップ/リストアを計画するときに考慮する必要のある事項を説明しますが、標準的な Red Brick、またはオペレーティング システムのマニュアルの代替となるものではありません。

バックアップ、およびリストア計画を決定する条件はいろいろありますが、その中でも次の項目を考慮する必要があります。

- ウェアハウスのサイズ
- ロードの頻度、および速度
- セグメント化の計画
- ロードの実行により影響を受けるセグメントの数
- バックアップの期間
- ウェアハウスの停止時間による、あきらかな損失
- 使用可能なハードウェア
- 包括的なクライアント操作の計画

十分に検討された包括的なバックアップ、およびリストアの計画は、どのようなデータベース アプリケーションにとっても不可欠な要素となります。標準的なデータウェアハウス アプリケーションの特徴、およびバックアップ/リストアの必要条件は、標準的な OLTP データベース アプリケーションとは異なります。バックアップ/リストア計画を単純にするデータウェアハウス アプリケーションの特徴には、以下が含まれます。

- 通常、データは、ほかのシステムから送られ、必要であれば再送されることが可能である。
- 通常、データは、クエリのため制御される時間内では更新されない。更新されるのは、ロードのため制御される時間内だけである。
- データベースが変更されると、その変更はすぐに取得されロード ファイルへ取り込まれる。

このような特徴が組み合わさると、以下のような状況が発生しバックアップ/リストア計画が複雑になる場合があります。

- データ ボリュームが大量になる。
- データ転送が遅くなる。
- デバイスの数が増えると、ハードウェア障害の可能性が高くなる。
- ネイティブ データベースのバックアップによる同時実行の問題が発生する可能性がある。

Red Brick のバックアップ/リストア

Red Brick の概要

Red Brick Warehouse は、ほかの多くのデータベース システムとは異なり、オペレーティング システムのファイル形式を使用してデータを保存します。これにより、オペレーティング システムのユーティリティ、またはサード パーティ製の高パフォーマンスなバックアップ/リストア ユーティリティを使用して、データベースとは異なる場所でバックアップ/リストア計画を実行することが可能となります。また、クエリ実行中にバックアップを同時に実行したり、DBA (データベース管理者) グループがバックアップ関連の作業をシステム管理者やオペレーション サポートのグループに委託することも可能となります。バックアップ/リストア計画のその他の作業も、Red Brick 固有の知識やデータベース権限がなくても実行、管理することができます。

バックアップ/リストアにより、障害によるデータの損失や損傷が発生した場合に、指定した状態までデータベースを復旧することができます。障害には、いろいろな種類があります。ディスク障害のようなハードウェアの問題により、データを損失することがあります。管理上のエラーにより、重要なファイルを上書きしたり削除したりすることがあります。ソフトウェアの問題により、誤ったデータがディスクに書き込まれることがあります。

Red Brick データベースは、多数のファイルで構成されていますが、ファイル内の情報は密接に関係付けられています。システム カタログには、データベースのレイアウト、および構造に関する情報が含まれ、インデックスにはデータに対するポインタが含まれています。また、参照整合性の制限により、テーブル内に同一のデータが保持されます。データベースが正しく処理されるには、このような相互に関係するデータの整合性が保証される必要があります。

バックアップ/リストアの概念を説明すると、バックアップとは、ある時点でのデータベースのスナップショットを取得すること、リストアとは、データベースをその時点の状態に戻すこと、を意味します。最後のバックアップ以降変更されていないファイルのバックアップや、置き換えるファイルと内容が同じファイルのリストアを省略することで、相当な時間を節約することができます。

Red Brick データベースは、データベース システム ファイル、およびデータとインデックスの物理格納ユニット (PSU) により構成されています。データベース システム ファイルの形式は RB_DEFAULT* で、このファイルはデータベースのホーム ディレクトリに格納されています。PSU の形式は、ユーザ定義、またはデフォルトを使用することができます。デフォルトの PSU 形式は dfltseg* で、ユーザ定義の PSU 形式はユーザが任意に定義することができます。

データベースへの書き込み処理が実行されると、データベース システム ファイル、およびデータベースの PSU の両方が更新されます。たとえば、新しい 5 つの列が CUSTOMER テーブルに追加されると、テーブルの PSU が更新され、新規データが反映されます。また、データベース システム ファイルは、テーブルの新規の列数などの情報により更新されます。

どの時点でも、システム ファイルと、データベースを構成する PSU のデータは一致している必要があります。

バックアップの種類

オンライン バックアップは、データベースの読み取り、および書き込み処理が可能な場合に実行されます。このようなバックアップは、Red Brick データベースのバージョンングが有効な場合のみ可能となります。

バックアップは、バージョンングが有効かどうかにかかわらず、データベースが読み取り専用 (クエリ専用) モードで使用されているときに実行することができます。バージョン 6.1 の新機能である LOCK DATABASE READ により、バックアップ中のデータベースは、かならず読み取り専用モードとなります。

バックアップには、主要なものとして完全バックアップとインクリメンタル バックアップ (差分バックアップ) の 2 種類があります。整合性のとれた状態への復旧を確実に実行するには、この 2 種類のバックアップを組み合わせることでデータベースをバックアップします。

完全バックアップは、データベース内のデータをすべてバックアップします。このようなバックアップは、通常はインクリメンタル バックアップよりも少ない頻度で実行されます。完全バックアップを実行するには、インクリメンタル バックアップよりも多くのデータ、および時間を必要とします。

インクリメンタル バックアップを実行すると、完全バックアップが最後に実行されてから変更された情報だけをバックアップすることができます。完全バックアップ、およびインクリメンタル バックアップの組み合わせの例としては、完全バックアップを毎月の初めに実行し、それ以降、インクリメンタル バックアップを次の完全バックアップまで毎週実行することが考えられます。

バックアップ計画を選択するには、注意が必要です。このことについては、このマニュアルで後述します。

バックアップ/リストア ツール

バックアップ/リストア計画を実行するツールは、数多く存在します。中には、ベンダー独自の特殊な機能を持つものもあります。このマニュアルでは、できるだけ一般的な情報を提供するために標準的な Red Brick、およびオペレーティング システム ツールのみを説明します。ただし、オペレーティング システムの場合、すべてのプラットフォームで使用されるツールが同じであるとは限りません。以下は、このマニュアルで説明するツールの一覧です。

- Red Brick の Table Management Utility (TMU : テーブル管理ユーティリティ)
- UNIX の dump/restore
- Windows NT/2000 の ntbackup.exe
- サード パーティ製のツール

バックアップ/リストア

Red Brick TMU によるバックアップ

データベースのサイズによっては、すべてのテーブルからデータをアンロードし出力ファイルを保存して、データの完全バックアップを実行する必要があります。アンロードされ、内部形式で保存されたデータは、テーブルに素早くロードし、ある時点の状態にリストアすることができます。この方法でバックアップを実行する場合、テーブルの DDL、およびロード スクリプトを保存することをお勧めします。

インクリメンタル バックアップを実行するには、テーブルをセグメント単位でアンロードします。最後のバックアップ以降のタイムスタンプが付いている PSU を含むセグメントをアンロードするだけで、インクリメンタル バックアップを実行することができます。

TMU によるデータのアンロードの詳細は、『Table Management Utility Reference Guide』を参照してください。

Red Brick TMU によるリストア

データベースを完全にリストアするには、DDL、およびロード スクリプトをテンポラリ ディレクトリにリストアします。DDL、およびロード スクリプトのリストアが完了したら、使用可能な DDL スクリプトで新規データベースの作成、および初期スキーマを再作成することができます。その後、ロード スクリプトを使用して、テーブルを作成します。

TMU を使用して、データベースの一部をリストアすることもできますが、データに対するセグメントを使用可能にしておく必要があります。この場合、セグメントを新規にアタッチするか、ALTER SEGMENT コマンドでクリアしたセグメントを再利用します。

バックアップが必要なファイルの決定

オペレーティング システムのレベルでバックアップ/リストアを実行するには、どのファイルをバックアップするか決定する必要があります。必要なファイルの完全な一覧を作成するには、次のクエリを実行します。

```
select physical_location from rbw_storage;
```

バックアップが必要なファイルの一覧は、ファイルのサイズ、および最新の更新日時と一緒に保存しておくと、リストアを実行するときに役に立ちます。

データベースの PSU 以外に、システム カタログ (RB_DEFAULT_* ファイル) をバックアップすることは、大変重要です。データベースの整合性を保持するには、リストア処理中にシステム カタログもリストアする必要があります。

UNIX dump ユーティリティによるバックアップ

UNIX システム上でデータベースを運用している場合、dump、および restore ユーティリティを使用してバックアップ計画を実行することができます。これらのユーティリティの特長は、使用するのが簡単で、オペレーティング システムの man のページで詳細が説明されていることです。さらに、これらのユーティリティを使用して、インクリメンタル バックアップを実行することも可能です。データベース管理者は、UNIX システム管理者と協力して、シェル スクリプトを作成したり、処理を自動化する cron などのツールを使用したりする方法を理解しておく必要があります。

dump ユーティリティには、ファイル システム全体のバックアップだけが可能である、という制限があります。不要なファイルをバックアップしないようにするには、Red Brick ファイルを専用のファイル システムに保存することをお勧めします。

インクリメンタル バックアップを実行するのは、非常に簡単です。最後のバックアップ以降の変更は、ユーティリティにより自動的に追跡されます。インクリメンタル バックアップを実行するときに含めるファイルとして、以下のいずれかを簡単に指定することができます。

- 最後のインクリメンタル バックアップ以降の変更分のみ
- 一連の差分
- 最後の完全バックアップ

dump/restore、および cron の詳細については、使用するオペレーティング システムの man のページを参照してください。

UNIX restore ユーティリティによるリストア

restore ユーティリティを使用して、ファイル システム全体、または特定のファイルとディレクトリをリストアすることができます。ファイルを指定するには、コマンドラインでファイル名の一覧を入力するか、リストアの実行中にその場でファイルを選択します。必要なファイルを復旧するには、関連するすべてのバックアップ ファイルに対し restore コマンドを実行する必要があります。

man のページでは、完全バックアップに対してリストアを実行した後で、一連のインクリメンタル バックアップに対して、復旧する必要がある時点までリストアを実行する方法について説明しています。この方法は、ファイルシステムの大部分、またはすべてをリストアする場合、あるいは処理するインクリメンタル バックアップ ファイルがほとんどない場合にもっとも効果的な方法となります。データベース内の変更の割合、および関連するインクリメンタル バックアップの回数によっては同じファイルを複数回リストアしてもかまいませんが、複数の中間的なバージョンのファイルをリストアすると、余計な I/O の量が増加する可能性があります。

restore ユーティリティは、ファイルをリストアするときに、ファイルの最初のタイムスタンプを使用しますが、これにより、データベースをリストアする、さらに効果的なアルゴリズムを検討することができます。もっとも古いバックアップから新しいバックアップに対して順番にリストアを実行するより、リストアを実行する前にどのファイルをリストアする必要があるのか再評価しながら、最新のバックアップからもっとも古いバックアップへリストアを実行することができます。ファイルがどのインクリメンタル バックアップにも存在しない場合、そのファイルは、最後に処理されたバックアップ、つまり完全バックアップからリストアされることとなります。

リストアが必要なファイルの一覧を作成するのは、難しい作業ではありません。バックアップからリストアする必要のあるファイルには、紛失ファイルと変更ファイルの 2 種類があります。紛失ファイルを検索するには、RBW_STORAGE に対してクエリを実行した結果と、UNIX の find コマンドでファイル システム上のファイルを検索し出力される現在のファイルの一覧を比較します。クエリの実行結果にはあってもフ

ファイル システムにはないファイルは、復旧が必要です。特定の日時以降、変更されたすべてのファイルの一覧を作成するには、UNIX の find コマンドを使用します。

このように、最後のバックアップから最初のバックアップへ順番にリストアを実行した場合、リストアが必要なファイルのみが、一度だけ書き込まれることになります。ファイルをバックアップからリストアするとき、そのタイムスタンプは置き換えられるファイルよりも古いものになるため、find コマンドで検索することができなくなります。これにより、ファイルの以前のコピーが現在の正しいコピーを上書きすることを防ぐことができます。

restore ユーティリティを使用すると、一連のファイルは現在の作業ディレクトリにリストアされます。restore ユーティリティを実行する前に、バックアップされたファイル システムを表すルートレベルのディレクトリにかならず移動しておいてください。

restore、および find の詳細については、使用するオペレーティング システムの man のページを参照してください。

Windows NT/2000 ntbackup.exe ユーティリティによるバックアップ

Windows NT、および Windows 2000 に実装されているバックアップ ユーティリティを使用して、対話式グラフィカル インターフェイス、またはバッチ ファイルやスケジューラ用のコマンド ラインからバックアップを実行することができます。このユーティリティで、インクリメンタル バックアップを実行することも可能です。

ntbackup.exe ユーティリティには、使用中のファイルをバックアップできないという、厳しい制限があります。そのため、Red Brick に対して実行するのが適当ではない場合があります。なぜなら、バックアップの処理中は、たとえ読み取り専用モードでも Red Brick をアクティブな状態にすることができなくなるからです。バックアップ中もデータベースを使用するには、そのような機能を持つサードパーティ製のツールを実行する必要があります。

ntbackup.exe ユーティリティの詳細については、ユーティリティのオンライン ヘルプを参照してください。

Windows NT/2000 ntbackup.exe ユーティリティによるリストア

このバックアップ ユーティリティは、対話式のグラフィカル インターフェイスでのリストアのみをサポートします。リストアの対象として、全体のディレクトリ構造、または個別のファイルを選択することができます。このユーティリティでリストアを実行する一般的な方法は、完全バックアップのすべての内容をリストアしてから、後続のすべてのインクリメンタル バックアップをリストアすることです。

前述の、UNIX 上で最後のものから最初のものへ順番に復旧していく方法は実行可能ですが、より難しくなります。リストアするファイルを決定するのは簡単かもしれませんが、バックアップ ファイルの一覧を手動で調べ復旧するファイルを検索する必要があります。複数のインクリメンタル バックアップを処理し、多数のファイルを復旧する必要がある場合、正しいバージョンのファイルをリストアするために、十分に注意する必要があります。

サードパーティ製のツールによるバックアップ

Veritas や Legato などのベンダーが開発したサードパーティ製のツールを使用して、Red Brick データベースのバックアップを実行することができます。

Red Brick は一般的なファイルを使用するため、サードパーティ製のツールが必要とする環境は最小限となります。膨大なボリュームのデータを処理する必要があるため、どのツールにも、インクリメンタルバックアップを実行する機能が最低限備わっています。その他に、以下の機能を持つツールが必要になることがあります。

- ビルトインのスケジューリング
- 複数のテープドライブへの書き込み
- ネットワークバックアップ機能
- 各バックアップの内容の自動追跡
- 特定の日時までリストアするファイルの自動選択

Windows NT、または 2000 で実行される製品の場合、ツールには以下の 2 つの項目がさらに必要となります。

- 使用中のファイルをバックアップする機能。読み取り専用でオンラインになっているデータベースのバックアップが必要な場合には、とくに重要な機能です。
- バッチモードによるリストアのサポート。

サードパーティ製のツールの詳細については、ベンダーが提供するマニュアルを参照してください。

バージョンングが有効な場合のバックアップ/リストア

確実に整合性のとれたバックアップの実行

バージョン管理されたデータベースの場合、データベース ファイルのスナップショットを取得する前に、バージョン ログが空である必要があります。バージョン ログが空であることを確認するには、DST_DATABASE に対してクエリを実行し、LATEST_MERGED_REVISION と CURRENT_REVISION が同じであることを確かめます。以下は、その例です。

```
select dbname from dst_databases

where dbname = 'AROMA'

and current_revision <> latest_merged_revision;
```

このクエリが 0 個の列を返した場合、データベース (この例では、AROMA データベース) をバックアップしても問題ありません。実行中のクエリが以前のバージョンにアクセスする場合、バージョン ログが空ではない可能性があります。このようなクエリは、READ_REVISION の最小の値を使用するクエリとして、DST_COMMANDS テーブルに格納されます。クエリが必要なければ、ADMIN データベースから中断することができます。

Red Brick Warehouse バージョン 6.1 では、2 つの新機能が追加され、この作業がより簡単になりました。LOCK DATABASE READ は、データベースに READ ロックを適用します。LOCK DATABASE を使用する上での注意点は、ロックはセッションと関連付けられるため、データベースをロックしたセッションが終了すると、ロックが解放されるということです。

ALTER DATABASE CLEAN VERSION LOG は、バージョン ログが空かどうかを確認します。しかし、セグメントが損傷しクリーニングが必要な場合、ステートメントは終了します。そのため、バージョン ログが空であることを確認するクエリを実行する前に、ALTER DATABASE CLEAN を実行する必要があります。

データベースの整合性が保持されている場合、バックアップ ユーティリティでバックアップを実行してもかまいません。

バージョンニングによるオンライン バックアップ

Red Brick バージョン 6.0.2 では、“凍結されるクエリ リビジョン”と呼ばれる新機能が追加されました。この機能により、特定のバージョンをユーザに表示させると同時に、管理者は、一般に公開する次のバージョンを準備することができます。次のバージョンが使用可能となるまで時間はかからないので、凍結されるバージョン以前のすべてのバージョンが削除される可能性は高くなります。また、凍結されるバージョンが使用されているとき、新規のデータがバージョン ログからデータベースに書き込まれることはありません。このように、LATEST_MERGED_REVISION と FROZEN_REVISION が同じ場合、凍結されるバージョンのディスク コピーを安全にバックアップすることができます。

凍結されるリビジョンをオンラインでバックアップするには、以下を実行します。

1. データベースを更新しないようにします。
2. 凍結するバージョンを作成します。
3. ログが空になり、LATEST_MERGED_REVISION が FROZEN_REVISION と同じになるまで待ちます。
4. システム カタログ (RB_DEFAULT_*) をバックアップします。
5. ユーザに、データベースへの完全なアクセスを許可します。
6. ほかのファイルをバックアップします。

整合性のない状態への復旧

データベースの一部を復旧する必要がある場合、データベースを整合性のない状態まで復旧してから Red Brick の CHECK TABLE コマンド、および TMU REORG を使用してデータベースを整合性が保持される状態にすることができます。

単一のテーブル、または少数のテーブルに対する PSU がない場合、整合性が保持されている状態まで復旧するには、最後のバックアップ以降に変更されたシステム カタログ、およびすべての PSU をリストアする必要があります。このような状況では、整合性のない状態までリストアした方が、時間が大幅に節約される場合があります。

この方法を実行するには、問題となる PSU のリストアだけを行います。現在のシステム カタログは、正しい場所に保存されています。この方法では、データベースのその他の PSU はシステム カタログと同期しているので、リストアする必要はありません。しかし、新規にリストアされる PSU は、現在のシステム カタログとは同期していません。新規の PSU とシステム カタログを同期させるには、CHECK TABLE を FIX オプションと一緒に実行し、テーブルのインデックスを再編成します。こうすると、現在のシステム カタログは、新規のテーブルの情報で更新されます。さらに、復旧されたテーブルから作成される集約テーブルも、再構築する必要があります。バージョン 6.1 では、TMU の REORG コマンドを使用して、テーブルのインデックス、および集約テーブルを 1 度の操作で再構築することができます。

この方法は、単一のテーブル、またはファクト テーブルに対し、大変効果的ですが、ディメンジョン テーブルを復旧する場合には、もっとも速い方法ではないかもしれません。ディメンジョン テーブルのプライマリキーは、ディメンジョン テーブルを参照するすべての STAR インデックスを強制的に再編成するからです。

復旧するデータ オブジェクトのスキーマが、最後のバックアップ以降に変更されている場合、この方法を実行することはできません。ALTER TABLE、および ALTER SEGMENT などの DDL コマンドは、この方法では実行不可能です。ロード、挿入、更新、および削除などの DML 操作は実行可能です。

概要

このマニュアルは、バックアップ/リストアの計画を実行するときに考慮する必要のある項目を説明しています。このような計画を実際に開発するのは、大変複雑な作業です。また、ウェアハウスのサイズ、設計、ロードの速度、バックアップの期間、および停止時間による損失と密接に関係しています。

バックアップ/リストア計画が複雑なため、計画を開発するための単純な手引書のようなものを作成することは不可能です。このことは、VLDB ウェアハウスの場合、特に当てはまります。