

## データベースにおけるバックアップ、リカバリ、および災害リカバリの意味

本書では、データベースの「バックアップ/リカバリ方式」を開発する方法、および「災害リカバリ方式」を開発する際に必要な要素について説明します。レプリケーション環境に関するデータベースのリカバリについては説明を省略しています。

### はじめに

バックアップ/リカバリの方式の開発は、いろいろな段階を経て遂行される作業です。これらの各段階を通して作業を遂行するためには、いくつかのデータベース用語、たとえばバックアップ/リカバリ、フル・データベース・バックアップとインクリメンタル・データベース・バックアップの違い、オンライン/オフライン/ライブの各バックアップの意味を理解しておく必要があります。

本書では、バックアップ/リカバリ計画を作成する各段階について簡単に説明し、例を1つ挙げています。データベースのリカバリにおいては時間がどれほど重要な要素となるかを説明します。また、バックアップしたデータベースをバックアップ・メディアからリストアできなかった場合にどうするのか、さらにどのようなデータベースの健全チェックを実行してデータベースとログのファイルの有効性を保証すべきなのかについても説明します。

データベースのバックアップとリカバリに使用する文を実例として示すため、データベースのバックアップとリカバリのコマンド例を記載しています。最後に、データベースを実行する実際のマシンが利用できなくなった状況に備えて、災害リカバリ方式を開発する場合の主要なポイントについて説明します。

### バックアップ、リカバリ、災害リカバリ

「バックアップ/リカバリ」と「災害リカバリ」は、まったく別のリカバリであると考えられるようになっていきます。

「バックアップ」とは、データベース・ファイルやログ・ファイルの内容をコピーするために使用するユーティリティ・プログラムです。データベース・ファイルは、データベース・ルート・ファイル、ログ・ファイル、ミラー・ログ・ファイル、および `dbspace` と呼ばれるその他のデータベース・ファイルから構成されています。

「リカバリ」とは、ある特定の時期にデータベースをリストアするために行う一連の作業です。リカバリは、ハードウェア障害またはメディア障害が発生した場合に実行されます。

ハードウェア障害とは、ディスク・ドライブ、コントローラ・カード、電源など、マシン内の物理コンポーネントの障害です。メディア障害とは、データを処理するときに発生した予期しないデータベース・エラーによる障害です。

リカバリを始める前に、障害のあるデータベースを習慣としてバックアップしておくことをお勧めします。障害のあるデータベースをバックアップすることにより、その状況を保存することができ、また安全な場所を確保することでファイルが誤って上書きされることを防止できます。さらに、リカバリ処理中に予期しないエラーが発生した場合に、Sybase サポート・センタがこれらのファイルを送ってもらうよう依頼することがあります。

「災害リカバリ」は、データベースのリカバリ事例とは異なるものです。災害リカバリの場合には、オペレーティング・システムや関連ソフトウェアのすべてをリカバリしなければ、いずれのデータベースのリカバリも開始することはできません。

データベースを構成するデータベース・ファイル

Adaptive Server Anywhere や SQL Anywhere のデータベースは、データを格納するディスク・ファイルから構成されています。Sybase Central の 'Create Database' または dbinit コマンドライン・ユーティリティを使用してデータベースを作成すると、メイン・データベース・ファイルすなわち **ルート・ファイル** が作成されます。デフォルトのデータベース・ファイルは database\_name.db という名前で作成されます。このメイン・データベース・ファイルには、データベース・テーブル、システム・テーブル、およびインデックスが含まれています。

その他のデータベース・ファイルはデータベースのサイズを拡張するものであり、dbspace と呼ばれています。dbspace には、テーブルとインデックスは含まれていますが、システム・テーブルは含まれていません。dbspace は、Sybase Central または Interactive SQL を使用し、CREATE DBSPACE コマンドを発行することによって作成することができます。デフォルトでは、dbspace ファイルは、drive:%path%dbspace\_name.db の書式で作成されます。データベースが1つ以上のdbspaceを使用しているかどうかは、Sybase Central 内の DBSpaces フォルダを開くことによって、あるいは sys.sysfile システム・テーブル(各 dbspace ごとに1行)を問い合わせることによって確認することができます。

**トランザクション・ログ**は、データベースの修正を記録するファイルです。データベースの修正としては、挿入、更新、削除、コミット、ロールバック、およびデータベースのスキーマ変更があります。トランザクション・ログは必須ではありませんが、使用することをお勧めします。データベース・エンジンはトランザクション・ログを使用して、最新のチェックポイントとシステム障害の間で行われた変更を適用します。チェックポイントは、コミットされたすべてのトランザクションがディスクに書き込まれることを保証します。dbinit というデータベース初期化ユーティリティを使用すると、デフォルトのファイル名が database\_name.log というログ・ファイルが作成されます。Sybase Central を使用してデータベースのログ・ファイルを作成すると、デフォルトのファイルは、

drive:¥path¥database\_name.log のようにログ・ファイルへの完全パスで表示されます。データベース・エンジンは、リカバリの間にこの場所でログ・ファイルを見つける必要があります。トランザクション・ログ・ファイルが明確に指定されていない場合、データベース・エンジンは、データベース・ファイルと同じディレクトリにログ・ファイルがあるものと想定します。

**ミラー・ログ**はオプション・ファイルであり、ファイル拡張子は mlg です。これは、トランザクション・ログのコピーであり、トランザクション・ログが使用できなくなった場合に、データの消失を特別に防止できるようにするものです。

## **オンライン・バックアップ、オフライン・バックアップ、およびライブ・バックアップ**

データベースのバックアップは、データベースが積極的にアクセスされているとき(**オンライン**)でも、あるいはデータベースがシャットダウンされているとき(**オフライン**)でも実行できます。データベースに対して通常のシャットダウン処理が行われると(この処理はキャンセルできない)、データベース・エンジンはデータベース・ファイルにデータをコミットします。

オンライン・データベース・バックアップは、コマンドラインで DBBACKUP を実行するか、Sybase Central の 'データベースのバックアップ' ユーティリティで実行できます。オンライン・バックアップ処理が開始されると、データベース・エンジンは、メモリに保持されたすべてのキャッシュ・データ・ページをディスク上のデータベース・ファイルに吐き出します。この処理はチェックポイントと呼ばれます。データベース・エンジンは、データベースのバックアップ中も、トランザクション・ログ・ファイルにアクティビティを記録する作業を続行します。ログ・ファイルは、バックアップ・ユーティリティがデータベースのバックアップを終了した後にバックアップされます。ログ・ファイルには、最後にデータベースがバックアップされてから以降に記録されたすべてのトランザクションが含まれます。このため、リカバリの間に、オンライン・フル・バックアップのログ・ファイルをデータベースに適用する必要があります。オフライン・バックアップのログ・ファイルは、リカバリ作業に適用する必要はありませんが、前のデータベースのバックアップを使用している場合には、リカバリで使用することができます。

**ライブ・バックアップ**は、DBBACKUP ユーティリティで -l コマンドライン・オプションを使用することで実行できます。ライブ・バックアップは、トランザクション・ログの重複コピーであり、プライマリ・データ・サーバのマシンが使用できなくなった場合に、セカンダリ・マシン上でシステムを再起動できるようにするものです。

## フル・データベース・バックアップとインクリメンタル・データベース・バックアップ

データベースのバックアップは、フルまたはインクリメンタルのいずれかのバックアップになります。フル・バックアップの場合、DBBACKUP と呼ばれるデータベースのバックアップ・ユーティリティがデータベースとログをコピーします。インクリメンタル・バックアップは DBBACKUP ユーティリティを使用して最後のフル・バックアップ以降のトランザクション・ログ・ファイルのコピーします。インクリメンタル・バックアップを実行すると、ミラー・ログのバックアップは行われません。ログ・ファイルをバックアップしてその名前を変更すると、トランザクションとミラーのログ・ファイル名が変更され、新しいログ・ファイルが作成されます。このため、手動によるミラー・ログのバックアップを計画する必要があります。バックアップ/リカバリ方式の計画を立てる場合には、このことに注意してください。

## バックアップ/リカバリ方式の開発

バックアップ/リカバリ方式を開発する場合の推奨手順を以下に示します。

ビジネスに対するバックアップ/リカバリの意味を理解する

経営者側がプロジェクトに対して時間とリソースを割り当てる

開発、テスト、時間測定、記録、健全チェック、展開、およびモニタを行う

リカバリに影響する外部の要因を認識する

セカンダリ・バックアップの問題に対処する

## ビジネスに対するバックアップ/リカバリの意味を理解する

企業データにアクセスしないで、ビジネスはどの程度生き延びることができるでしょうか？この答えを分、時間、または日数で表してみてください。

リカバリ時間が分単位であれば、データベースのバックアップ/リカバリはビジネスのニーズにとって不可欠なものであり、何らかのバックアップ/リカバリ方式を実装することが必須となります。リカバリが時間単位であれば、より長い時間をかけてこの作業を実行することができます。リカバリを日数単位で表せるような場合、データベースをリカバリしなければならないという必要性は変わりませんが、時間は重要な要因にはならないと考えられます。

## 経営者側がプロジェクトに対して時間とリソースを割り当てる

経営者側は、開発のための財源を決定し、またバックアップ/リカバリ方式の実装を決定しなければなりません。この方式は、会社のビジネス・ニーズに応じて、基本的な方式の場合もあればきわめて大規模な方式の場合もあります。バックアップ/リカバリ方式を開発した後、予測されるバックアップ/リカバリの時間を経営者側に通知する必要があります。時間的な問題に対処するために、経営者側には代替りの解決策が用意されています。これら代替の解決策としては、ハードウェアを追加で要求する、バックアップ・メディアを改良する、バックアップ・スケジュールを変更する、より長いリカバリ時間/バックアップ時間を受け入れる、などが考えられます。その後、企業のニーズにふさわしい解決策を決定することは経営者側の責任になります。

## 開発、テスト、時間測定、記録、健全チェック、展開、およびモニタを行う

これらの各段階は、バックアップ/リカバリ方式を開発する上での中核となるものです。バックアップ/リカバリのコマンドを作成します。これらのコマンドが設計どおりに機能することを確認します。フルまたはインクリメンタルのオンライン・バックアップが正常に動作しますか？ コマンドが希望の結果を生み出すことを確認します。

バックアップ/リカバリのコマンドを実行して時間を推定することにより、これらの作業にかかる時間の程度を感覚的に把握できるようになります。この情報を利用して、どのコマンドをいつ実行すべきかを判断します。

バックアップのコマンドを記録し、バックアップが保持されている場所を記した手順書を作成し、さらに使用する命名規則と実行するバックアップの種類を特定します。この情報は、個人がバックアップを検査したりデータベースのリカバリを実行しなければならないとき、かつデータベース管理者(DBA)の支援を得られないときにきわめて重要となります。

バックアップ手順の中に健全チェックを組み込みます。データベースが破壊されていないことを保証するためには、データベースを検査する必要があります。データベースの健全チェックはデータベースをバックアップする前に実行でき、あるいはバックアップから取り出したデータベースのコピーに対して実行できます。

バックアップ/リカバリを展開するには、バックアップ手順を運用サーバにセットアップすることが必要です。必要なハードウェアが正しく配置されていることを確認し、また作業を実施するのに必要なサポート・ソフトウェアが備わっていることを確認します。手順を修正して、環境の変化を反映するようにします。ユーザ ID、パスワード、サーバ名、およびデータベース名を変更して環境の変化に対応します。

予期せぬエラーを防止するためバックアップ手順をモニタします。処理内で行った変更がいずれもマニュアルに反映されていることを確認します。

## リカバリに影響する外部の要因を認識する

データベースのリカバリに影響する外部の要因としては、時間、ハードウェア、およびソフトウェアがあります。実行すべきその他の作業を入力するためのリカバリ時間を追加できるようにします。これらの作業は、リカバリ・コマンドの入力やテープの出し入れと同程度に簡単に実行できます。時間に影響する要因は、データベース・ファイルのサイズ、リカバリ・メディア、ディスク領域、および予期しないエラーです。データベースで DBSPACE を使用して、バックアップからのリストアに失敗した場合、バックアップは無効になります。リカバリに追加するファイルの数が多いほど、リカバリの失敗が許される事例が増加することになります。たとえば、4つの DBSPACES、メイン・データベース・ファイル、ミラー、およびトランザクション・ログ・ファイルを使用するものとします。ログ・ファイルの1つをテープ・バックアップからリストアすることに失敗した場合、別のログ・ファイルを使用します。DBSPACE のバックアップをディスクにリストアすることに失敗した場合、前のフル・データベース・バックアップを復元する必要があります。バックアップ/リカバリ方式は発展するものであるため、機器とソフトウェアのパフォーマンスを検査して、パフォーマンスが期待どおりであることを確認する必要があります。

## 健全チェックを実施してデータベースのバックアップを保護する

データベースの健全チェックは、データベース・ファイルとログ・ファイルに対して実行されるものであり、各ファイルが破壊されていないことを確認します。DBVALID と呼ばれるデータベースの有効性ユーティリティを使用すると、あらゆるテーブルの各レコードをスキャンし、またテーブル上の各インデックス内の各レコードを調べることができます。データベース・ファイルが破壊されている場合、以前のデータベースのバックアップからリカバリしなければなりません。データベースは、バックアップする前に検証することができ、あるいはバックアップから取り出したデータベースのコピーに対して検証することができます。

DBVALID を使うとデータベースの検査に時間がかかる場合があります。このため、いつ DBVALID を実行するかを決定する必要があります。フル・データベース・バックアップを実行する前にデータベースの有効性チェックを行う場合は、バックアップする前のデータベースにデータベースのポインタ・エラーがないことを確認します。バックアップ・メディアからリストアしたコピーに対して DBVALID を実行する場合は、以下のチェックを実施します。

バックアップ・メディアからデータベースを正常にリストアできることを確認します。運用環境に影響を及ぼすことなく、データベースに対してデータベース有効性チェックを実行できるようになります。

DBVALID を非運用マシン上で実行すると、運用環境への影響をなくすることができます。検証処理で使用するデータベースはリカバリに適用することはできません。データベースがいったん開始されるとログ・オフセットが変更されるからです。

ログ・ファイルの破壊の有無は、DBTRANユーティリティを使用して検査します。DBTRANユーティリティはログ・ファイルを SQL ファイルに変換し、ログ・ファイルが破壊されていないことを確認します。このことは、大規模なデータベースのリカバリが必要で複数のログ・ファイルを適用しなければならないときに重要となります。データベースのリカバリ中にログ・ファイルの破壊が見つかったりリカバリは停止されるので、その時点から障害の時点までのリカバリを手動で行えるかどうかを判断する必要があります。ミラー・ログをバックアップしている場合は、破壊したトランザクション・ログをバックアップのミラー・ログの内容に置き換えて、そのログ・ファイルを利用することができます。別の方法としては、ログ・ファイルが以前に変換されている場合、その変換された SQL ファイルを、残りの変換済み SQL ファイルとともにデータベース内に読み込むことができます。ログ・ファイルを利用する代わりに SQL ファイルをデータベースに読み込むという方法はリカバリの別の選択肢になりますが、この方法は、データベースがレプリケーションに加わっている場合には使用できないことに注意してください。

## バックアップ / リカバリ方式の例

ここで、'Data Online'という架空の会社のバックアップ / リカバリ方式を開発してみましょう。Data Online は、就業時間中にデータベースに格納された企業データにアクセスできることを必要としています。就業時間は月曜日から金曜日までの午前 8 時 ~ 午後 8 時までです。就業時間中、2 時間以上この情報が利用できなくなると、Data Online は仕事を続行できません。就業時間外に企業データにアクセスできることは必須ではありません。

データベースはバックアップされていません。データベースのサイズは 600MB であり、400MB のメイン・データベース・ファイルと、2 つの 100MB データベース領域(それぞれ複数のテーブルとインデックスを含む)から構成されています。データベースはトランザクション・ログとミラー・ログを使用しています。日々の業務活動によって、ログ・ファイルは 1 日当たり 80MB の割合で増大します。以上で、データベースのバックアップとリカバリにかかる時間の基準値を作成することが可能です。時間測定値は架空の値です。実際には、マシン、データベースのサイズ、マシン上で動作するその他のソフトウェアなどによって値は変化します。

データベースのバックアップ・ユーティリティの名前は DBBACKUP です。これを使用するとデータベース・ファイルとログ・ファイルをバックアップできます。DBBACKUP を使用してフル・データベース・バックアップを行うと、メイン・データベース・ファイル、データベース領域、およびトランザクション・ログ・ファイルのコピーが作成されます。

## バックアップとリカバリの基本操作の時間測定

フル・データベース・バックアップのテストを数回実行した後、平均バックアップ時間を計算すると 30 分になりました。インクリメンタル・バックアップについては、増大分を取り除くため、ログ・ファイルの名前を変更してから再起動することも必要になります。ここで、インクリメンタル・バックアップについて複数回のテストを実行しました。このテストでは、トランザクションとミラーのログ・ファイルをバックアップすることで、データベースに対する変更だけをバックアップしています。複数回のバックアップ・テストを実施した後、平均インクリメンタル・バックアップ時間を計算すると、40MB のサイズのログ・ファイルについて 10 分になりました。Data Online は、ディスク・ファイルをテープにバックアップするサード・パーティのソフトウェア・ツールを持っています。データベースのバックアップはすべてテープにコピーします。データベースとログについて平均バックアップ時間とテープからの平均リストア時間を計算すると、インクリメンタル・データベース・バックアップの場合にそれぞれ 15 分と 5 分でした。これで、以上の時間測定値を使用すればリカバリ方式を開発することができます。

## 障害からの総リカバリ時間を推定する

以下の条件を仮定すれば、前節で推定した時間を用いてリカバリ時間を推定できるようになります。

バックアップからデータベースのリカバリが必要な場合、テープからディスクへのデータベースのリストアに 15 分かかります。

次に、日々の 2 つのインクリメンタル・データベース・バックアップについて新たに 10 分が追加されます(各ログ・ファイルをテープからリストアするのに 5 分)。

現在のトランザクション・ログ・ファイルは破壊されておらず、データベースに適用できるものと想定します。リストアしたデータベースにログを適用するというリカバリ処理に約 5 分かかります。

推定リカバリ時間は、以下に示すように合計で 55 分と計算されます。

バックアップ・メディアからのリカバリ段階

### 操作時間

フル・データベース・バックアップ 15 分

インクリメンタル・データベース・バックアップ 10 分(各 5 分の 2 倍)

バックアップ・メディアの小計時間 25 分

データベースのリカバリ段階(リカバリしたデータベースにインクリメンタル・データベース・バックアップを適用)

### 操作時間

3 つのログ・ファイルの適用 15 分(各 5 分の 3 倍)



コマンドの発行 15分

データベースのリカバリの小計時間 30分

総リカバリ時間：55分

このリカバリ例では、データベースのリカバリを実行して3つのインクリメンタル・バックアップを適用するのに約1時間かかることがわかります。夕方にフル・データベース・バックアップを実行するとともに、翌日に2つのインクリメンタル・バックアップを実行すれば、翌就業日が終了するまでにデータベースのリカバリは完了し、2時間以内にデータベースのリカバ리를完了するという業務目標を達成することができます。したがって、フル・データベース・バックアップを毎夜実施し、2つのインクリメンタル・データベース・バックアップを各就業日の正午に実施し、もう1つは午後6時に実施することになります。5週間分のデータベースのバックアップをテープに保持することをお勧めします。

以上で、バックアップ/リカバリ方式を経営者側に提示できます。

## リカバリ中にエラーが発生したときの対処

フル・データベース・バックアップのリストア処理中に、テープからデータベース領域をリストアすることに失敗することがあります。このデータベースのバックアップ・コピーは無効になります。この場合、利用可能な次のフル・データベース・バックアップ、およびデータベースを障害時点のオリジナル・データベースにリカバリするのに必要なインクリメンタル・バックアップを指定します。この例では、前夜のフル・データベース・バックアップがあるのでこれでリカバリ処理を開始できます。前日のフル・データベース・バックアップを使用してリカバリ処理を開始します。リカバリ時間は、前日のファイルをリストアするために、新たに25分が加算されることになります。また、6つのログ・ファイルを適用するのに、さらに60分が加算されます(4つのインクリメンタル・データベース・バックアップ、および前日のフル・データベース・バックアップと失敗したフル・データベース・バックアップに伴う2つのトランザクション・ログ・ファイル)。

前日のバックアップからのリカバリ

メディアからのリカバリ段階。最初の試みで、テープからフル・データベース・バックアップを正常にリストアすることに失敗

### 操作時間

フル・データベース・バックアップ 15分

インクリメンタル・データベース・バックアップ 10分(各5分の2倍)

メディアのリカバリの小計時間 25分

2回目の試みで、前日のフル・データベース・バックアップから正常にリストア

## 操作時間

フル・データベース・バックアップ 15分

インクリメンタル・データベース・バックアップ 10分(各 5分の 2倍)

メディアのリカバリの小計時間 25分

データベースのリカバリ段階(リカバリしたデータベースにインクリメンタル・データベース・バックアップを適用)

## 操作時間

リカバリしたデータベースに 6つのログを適用 30分

コマンドの発行 30分

総リカバリ時間：1時間 50分

この 2番目の事例は、予期しない出来事に対して準備をしておく必要のあることを示しています。あらゆるデータベースのリカバリは、前のリカバリとは異なるものです。フル・データベース・バックアップからリストアしたデータベース・ファイルの 1つが不良の場合、ログ・ファイルが有効であることを確認してください。有効であれば、少なくとも前日のフル・データベース・バックアップをリカバリし、ログを適用することでこの故障時点以降をリカバリできるということです。

## リカバリに関するその他の情報

どのようなデータベースのリカバリを実行する場合でも、実行前に障害のあるデータベースをバックアップするようにしてください。その方法がわからない場合や自信のない場合は、Sybase サポート・センタにお問い合わせください。データベースのリカバリ方法がすでにわかっている場合でも、サポート・センタに連絡することで別のアイデアを得ることができ、また作成したリカバリ計画を Sybase サポート・センタによって検証してもらうことができます。

複数のフル・データベース・バックアップによって、リカバリでの複数のフェールセーフ・ポイントに対処できることに留意してください。各インクリメンタル・バックアップに対して DBTRAN を実行し、ログ・ファイルが有効であることを確認します。ログ・ファイルを適用する代わりに変換済みの SQL ファイルを使用する方法がありますが、システムがレプリケーションに参与している場合には、この方法を使用しないようにしてください。これは、レプリケーションを破壊することになります。変換済みの SQL ファイルを使用した場合、残りのログ・ファイルをすべて変換してデータベースに読み込ませる必要があります。このリカバリ方法は、ログ・ファイルを適用するよりも多くの時間がかかります。

## セカンダリ・バックアップの問題

各バックアップについて、バックアップ保存期限を定義します。

バックアップ・メディアがテープの場合、廃棄までの使用回数を決定します。

テープからのリストアを実施してリストアが可能であることを確認することで、バックアップ手順をテストします。

別のリカバリ例を作成し、これを用いてデータベースのリカバリを実施します。

リカバリ手順が最新状態であることを確認します。

## バックアップとリカバリのコマンド

この節で使用するデータベース・サーバは、Adaptive Server Anywhere 6.0です。SQL Anywhere Version 5.x が現在のデータベース・ソフトウェアのバージョンである場合、パーソナル・サーバ **dbeng6** を **dbeng50** に、あるいはネットワーク・サーバ **dbsrv6** を **dbsrv50** に交換してください。次に示す例では、パスは示されていません。状況に応じてインストールのデフォルト・パスを使用する場合もあれば、パスを変更する場合もあるからです。その他のユーティリティは、特に指定しない限り、データベースのバージョン間での違いはありません。

この例で使用するデータベース・サーバは、Adaptive Server Anywhere 6.0 パーソナル・サーバ **dbeng6** です。SQL Anywhere の場合、スタンドアロン・エンジン **dbeng50** になります。

## コマンドのテスト

以下のコマンドは、このままバックアップやリカバリのスクリプトにコピーして使用することを想定したものではありません。コマンドを実際に使用する前に、必ずテストを繰り返し実行してください。

## 使用するディレクトリ

この例で使用するファイル・ディレクトリ

- C:¥prod (運用データベースのディレクトリ)
- C:¥bkdb (障害データベースのディレクトリ)
- D:¥rcv (リカバリ・データベースのディレクトリ)
- D:¥dbbkup (データベースのフル・バックアップ)
- D:¥lgbkup (ログのインクリメンタル・バックアップ)
- ¥HHhhhhh0800 (午前 8:00 のバックアップ)
- ¥HHhhhhh1200 (午前 12:00 のバックアップ)
- ¥HHhhhhh1700 (午後 5:00 のバックアップ)

この例では、データベースはパーソナル・サーバにマウントされています。

DBENG6 -n server C:\prod\database.db

## バックアップの作成

データベース・ファイルとログ・ファイルについて、オンラインのフル・バックアップを作成します。

```
DBBACKUP -c "uid=dba;pwd=sql;eng=server;dbn=database" D:¥dbbkup
```

トランザクション・ログ・ファイルとログ・ファイルについてのオンラインのインクリメンタル・バックアップが午前 8:00 にトランケートされます。

```
DBBACKUP -t -x -c "uid=dba;pwd=sql;eng=server;dbn=database" D:¥lgbk¥h0800
```

トランザクション・ログ・ファイルとログ・ファイルについてのオンラインのインクリメンタル・バックアップが午前 12:00 にトランケートされます。

```
DBBACKUP -t -x -c "uid=dba;pwd=sql;eng=server;dbn=database" D:¥lgbk¥h1200
```

トランザクション・ログ・ファイルとログ・ファイルについてのオンラインのインクリメンタル・バックアップが午前 8:00 にトランケートされます。

```
DBBACKUP -t -x -c "uid=dba;pwd=sql;eng=server;dbn=database" D:¥lgbk¥h1700
```

## バックアップのリカバリ

フル・バックアップからデータベースをリカバリし、インクリメンタル・バックアップ(トランザクション・ログ)のすべてと障害時点でのログ・ファイルを適用して現時点へデータベースをリカバリするには、以下の操作を行います。

- 1 障害のあるデータベースとログのバックアップ・コピーを **c:¥bkdb** に作成します。
- 2 リカバリ・データベースを **d:¥rcv** に作成します。
- 3 フル・バックアップによるデータベースだけを **d:¥rcv** に作成します。

-a (ログの適用)スイッチを使用してログ・ファイルを適用することで、フル・データベース・バックアップを開始します。サーバのウィンドウをモニタしてエラー・メッセージを確認します。リカバリが進行中であることを示すメッセージが表示されるはずですが。

```
DBENG6 D:¥rcv¥database.db -a D:¥dbbkup¥database.log
```

残りのログ・ファイルを適用します。

```
DBENG6 D:¥rcv¥database.db -a D:¥lgbk¥h0800¥database.log
```

```
DBENG6 D:¥rcv¥database.db -a D:¥lgbk¥h1200¥database.log
```

```
DBENG6 D:¥rcv¥database.db -a D:¥lgbk¥h1700¥database.log
```

ここで、データベース障害時のログ・ファイルを適用します。

```
DBENG6 D:¥rcv¥database.db -a D:¥bkdb¥database.log
```

データベース・ファイルだけを運用ディレクトリにコピーします。新しいログ・ファイルが作成されます。これで、データベースに対して健全チェックを実行する準備が完了です。

## 健全チェックを実施する

ここで、パーソナル・サーバ(dbeng6)またはスタンドアロンのデータベース・エンジン(dbeng50)を使用してデータベースを開始します。これは、ユーザがデータベースに接続し、データベースが有効であることを確信できるようにするためです。

データベースはパーソナル・サーバにマウントされます。

```
DBENG6 -n server C:¥prod¥database.db
```

DBVALID を実行してデータベースが破壊されていないことを確認します。

```
DBVALID -c "uid=dba;pwd=sql;eng=server;dbn=database"
```

## リカバリ後のバックアップを作成する

バックアップ・ディレクトリにあるデータベースの古いコピーをすべて除去します。特定のバックアップ・メディアにファイルをバックアップすることもできます。

データベース・ファイルとログ・ファイルのリカバリ後のバックアップを作成します。

```
DBBACKUP -c "uid=dba;pwd=sql;eng=server;dbn=database" D:¥dbbkup
```

## ユーザがデータベースにアクセスできるようにする

スタンドアロンまたはパーソナル・サーバのデータベース・エンジンを使用して健全チェックを実施し、データベースの有効性を確認し、さらにデータベースのリカバリ後のバックアップを実施したので、ここで、データベースをシャットダウンします。リカバリの前に通常の方法でデータベース・エンジンを開始します。データベース・エンジン、およびデータベースに対するユーザのアクティビティをモニタし、予期しないエラー・メッセージがないことを確認します。

## 災害リカバリ

災害リカバリがデータベースのリカバリと異なる点は、通常、マシンが利用できなくなるという点です。これは、洪水、自然災害、あるいは動作不能のマシンによるものです。このような状況では、オペレーティング・システム、システム・ソフトウェア、データベース・ソフトウェア、およびアプリケーション・ソフトウェアを物理的に異なるマシンにリカバリすることが必要となります。このマシンは、類似のマシンの場合もあれば同一のマシンの場合もあります。マシンが異なる場合、災害リカバリに影響する可能性があります。システム・ソフトウェアとデータベース・ソフトウェアをインストールすれば、データベースのリカバリ手順を開始できます。

## 災害リカバリ計画に関するその他の情報

以下に、災害リカバリ方式を開発する際に必要となるポイントを一覧で示します。これらのポイントについて、計画した方式を評価し、見落としているポイントがあれば必要に応じて方式を更

新するようにしてください。この一覧は、必要と思われるポイントの代表例にすぎず、実際には各サイトのハードウェア、ソフトウェア、所在地、スタッフの構成によって異なる可能性があります。

- 互換性のあるシステムとネットワーク・ハードウェアを備えたオフサイトの場所(サイトから離れた場所)を確保します。
- オフサイトのバックアップを取り出すための手順を確立します。
- 災害リカバリ手順は複数の箇所で保持するようにします。オフサイトの場所には、オペレーティング・システム、システム・ソフトウェア、データベース、およびアプリケーション・ソフトウェアの各バックアップ、さらには災害リカバリの作業に携わる人物も用意しておく必要があります。
- 災害リカバリの定期的なテストを計画し、各テストの後にその結果を評価します。
- 災害リカバリ手順に、オペレーティング・システム、システム・ソフトウェア、データベース、およびアプリケーション・ソフトウェア用の、すべてのインストール・ソフトウェアとそのパスワードのコピーが含まれていることを確認します。
- すべてのソフトウェア・ベンダの名前、電話番号、製品サポート計画番号、およびソフトウェア・ライセンス情報を災害リカバリ手順に添付します。
- 災害リカバリ手順を完了するのに要する合計時間を記録します。
- 災害リカバリ手順を最新状態に保ち、オフサイトに保持されたコピーを更新します。

## まとめ

ここに示した資料は、データベースのバックアップ/リカバリ方式を開発して実装する場合の基本的な情報を提供するものです。この資料でわれわれは、データベースに使用したファイルを特定し、オンライン、オフライン、およびライブの各バックアップの違いについて明らかにしました。投資を保護するためには、健全チェックをバックアップ/リカバリ方式に組み込むことの重要性を強調しました。バックアップとリカバリの例を挙げてバックアップ/リカバリ方式の開発に必要となる要因を説明しました。また災害リカバリに伴う作業について説明しました。災害リカバリはおろそかにすべきではありません。機器に発生する不慮の事故によってサーバやデータベースが動作不能になる可能性があります。さらに強調したいことは、バックアップとリカバリの手順は繰り返してテストするということです。思いこみをなくし、すべてが想定どおりに機能することを確認します。われわれは、ここで説明した内容がバックアップ/リカバリ方式と災害リカバリ方式の開発と実装に役立つことを願っています。

バックアップとリカバリの詳細については、『 Adaptive Server Anywhere ユーザーズ・ガイド』および『 Sybase SQL Anywhere ユーザーズ・ガイド第 1 巻』の「バックアップとデータ・リカバリ」という章を参照してください。

