Db2 バックアップ・リカバリー手順

発行日：2025/11

日本アイ・ビー・エム システムズ・エンジニアリング株式会社

バックアップ・リカバリーの手順

はじめに

当資料の構成

１．オフライン・バックアップを使用したリカバリー

２．オンライン・バックアップを使用したリカバリー

３．表スペース・バックアップを使用したリカバリー

４．増分バックアップを使用したリカバリー

５．RECOVERコマンドを使用したリカバリー

はじめに

当資料では、Db2 12.1 におけるバックアップ・リカバリーに関する手順を紹介します。

対象とする読者

- Db2 12.1 でバックアップ・リカバリーの計画を担当されている方
- Db2 12.1 でバックアップ・リカバリーの実施を担当される方

対象製品

- Db2 12.1 for Linux, UNIX and Windows

シナリオで使用するデータベース

- 定義されているデータベース
  - サンプルデータベース(SAMPLE)
- 定義されている表スペース
  - カタログ表スペース（SYSCATSPACE）
  - 一時表スペース（TEMPSPACE1）
  - ユーザー表スペース（USERSPACE1）

当資料の構成

当資料では、以下のリカバリー手順を記載します。
- １．オフライン・バックアップを使用したリカバリー
- ２．オンライン・バックアップを使用したリカバリー
- ３．表スペース・バックアップを使用したリカバリー
- ４．増分バックアップを使用したリカバリー
- ５．RECOVERコマンドを使用したリカバリー

各章の構成
１章から５章までの各章、以下の構成をとっています。
- バックアップ・リカバリーに関する概要
- リカバリーまでのシナリオ
- 手順

手順のコマンドや出力例はDb2 12.1.2での実行結果を添付しています。

アーカイブロギングが必要な環境では　以下のコマンド例を実行します。

```
$ db2 update db cfg for SAMPLE using logarchmeth1 disk:/work/log/arclog
DB20000I  UPDATE DATABASE CONFIGURATION
コマンドが正常に完了しました。
```

オフライン・バックアップを使用したリカバリー

01

1. オフライン・バックアップを使用したリカバリー

1-1. オフライン・バックアップ取得時点にリカバリーする

1-2. オフライン・バックアップ取得時点から最後のログまでリカバリーする

1-3. オフライン・バックアップ取得時点以降の任意の時点にリカバリーする

1-1. オフライン・バックアップ取得時点にリカバリーする

内容

– オフライン・バックアップを使用したリカバリーの手順

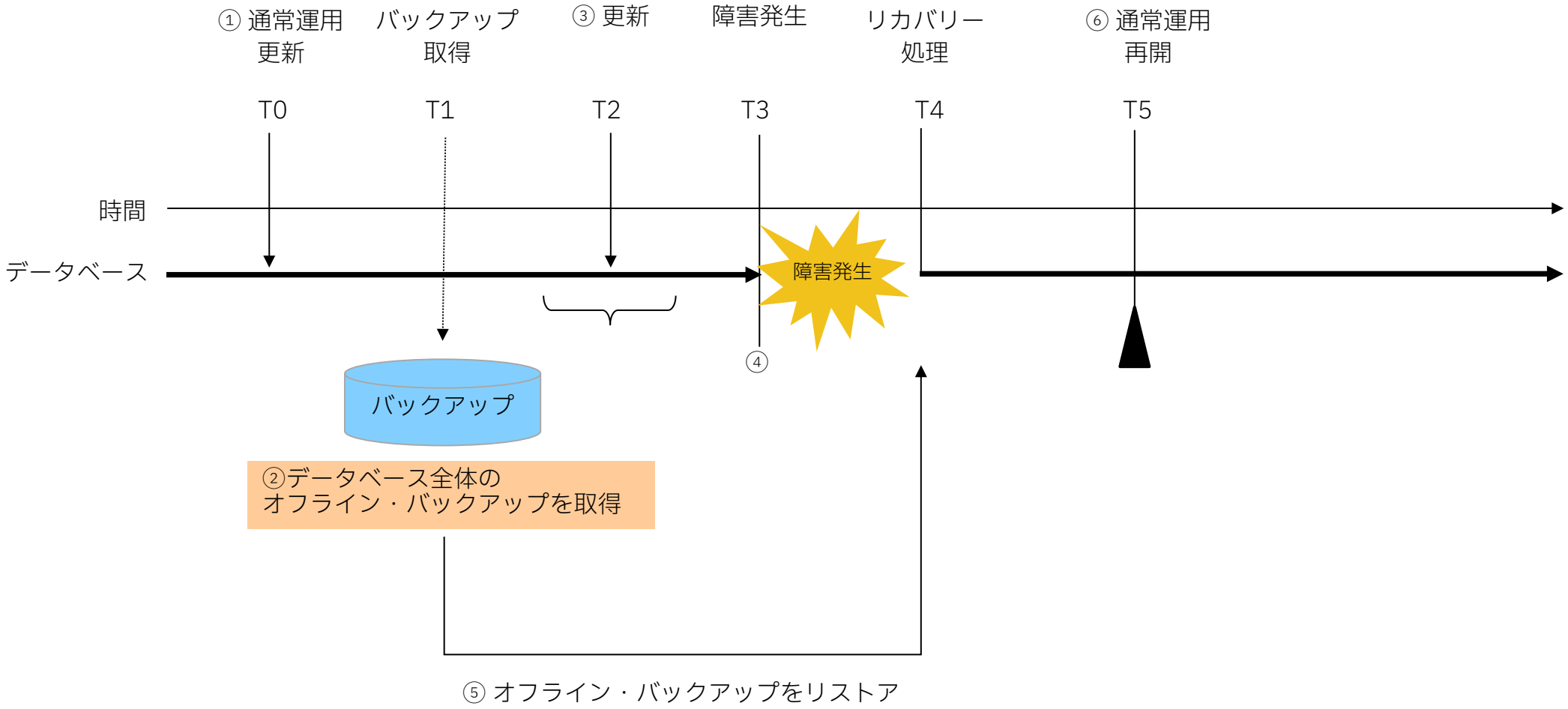
概要

– データベース全体のオフライン・バックアップを使用してリストアを行い、オフライン・バックアップ取得時点までデータベースをリカバリーする手順を紹介する。

手順

– データベースのオフライン・バックアップを取得する。
– 取得したバックアップイメージを用いて、リストアする。
– データベースに接続し、バックアップ取得時点までリカバリーされたことを確認する。

1-1. リカバリーのシナリオ (1/2)



1-1. リカバリーのシナリオ (2/2)

	シナリオ	時刻
①	データベースに対して更新を行う。	T0
②	データベースをオフライン状態にし、オフライン・バックアップを取得する。	T1
③	データベースに対して更新を行う。	T2
④	データベースに障害が発生し、データベースが使用不能になったことを想定。	T3
⑤	②で取得したデータベースのオフライン・バックアップをリストアする。	T4
⑥	データベースは、オフライン・バックアップを取得した②の時点にリカバリーする。	T5

1-1. 手順 (1/2)

- データベースに対して更新を行う。（①）

```
$ db2 "insert into TAB1 values ( 1, 'TEST' )"
DB20000I  SQL コマンドが正常に完了しました。
$ db2 "select * from TAB1"
C1          C2
----------- ----------
          1 TEST
  1 レコードが選択されました。
```

- データベース全体のオフライン・バックアップを取得する。（②）

```
$ db2 backup db SAMPLE to /work
バックアップは成功しました。このバックアップ・イメージのタイム・スタンプは  20250807095328 です。
```

- データベースに対して更新を行う。（③）

```
$ db2 "insert into TAB1 values ( 2, 'TEST' )"
DB20000I  SQL コマンドが正常に完了しました。
$ db2 "select * from TAB1"
C1          C2
----------- ----------
          1 TEST
          2 TEST
  2 レコードが選択されました。
```

1-1. 手順 (2/2)

- ここで、データベースに障害が発生し、データベースが使用不能になったことを想定する。（④）
- ②で取得したデータベース全体のオフライン・バックアップをリストアする。（⑤）

```
$ db2 restore db SAMPLE from /work taken at 20250807095328
SQL2539W  リストアするバックアップ・イメージに指定された
名前がターゲット・データベースの名前と同じになっていま
す。 バックアップ・イメージ・データベースと同じ既存デ
ータベースをリストアすると、現行のデータベースがバック
アップ・バージョンによって上書きされます。
続けますか。 (y/n) y
DB20000I  RESTORE DATABASE コマンドが正常に完了しました。
```

- データベースに接続し、表TAB1の内容を確認する。③の更新が反映されておらず、②の状態に正常にリカバリーしたことがわかる。

```
$ db2 "select * from TAB1"
C1          C2
----------- ----------
          1 TEST

  1 レコードが選択されました。
```

1-2. オフライン・バックアップ取得時点から最後のログまでリカバリーする

内容

– オフライン・バックアップを使用したリカバリーの手順

概要

– データベースのオフライン・バックアップと、バックアップ取得後のアーカイブ・ログを最後まで使用して、データベースをリカバリーする手順を紹介する。

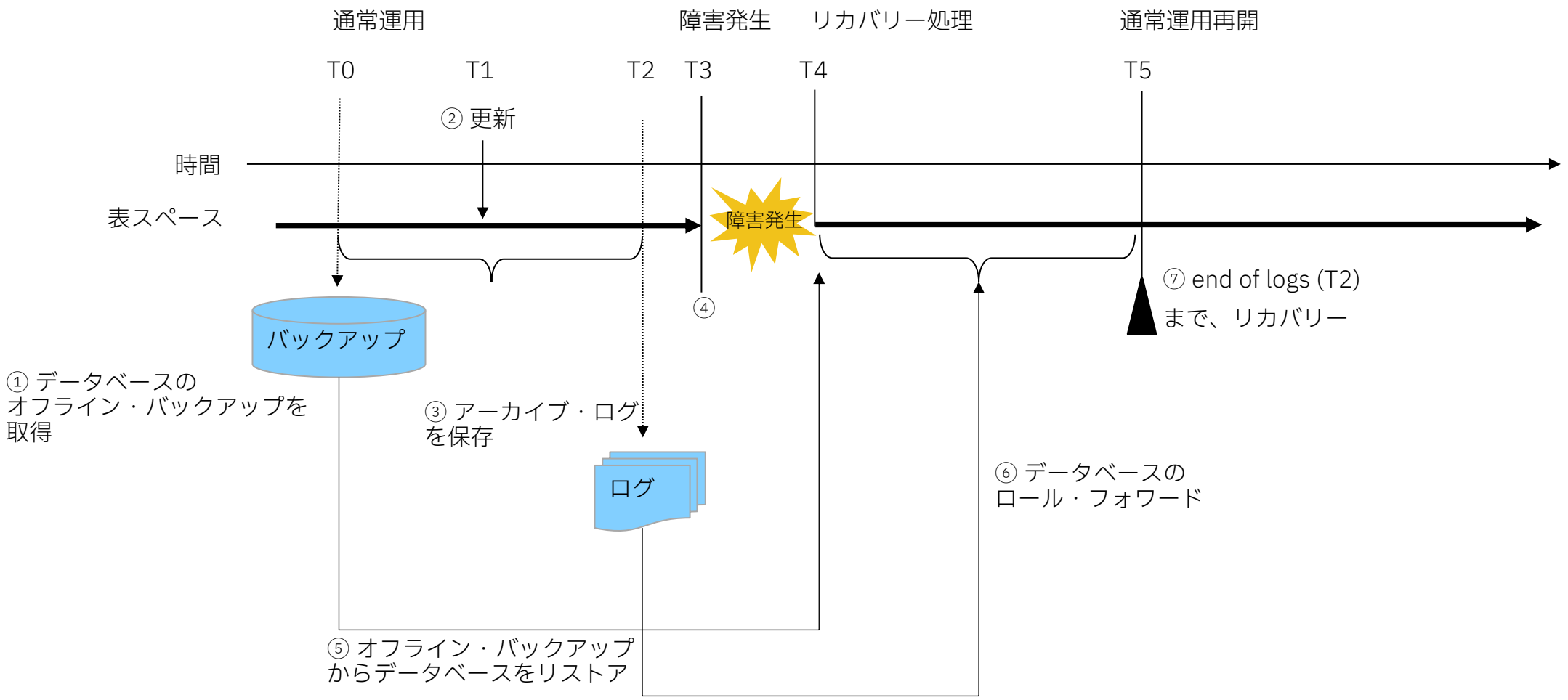
前提

– ロールフォワード・リカバリーを利用するための、ロギングのタイプをアーカイブ・ロギングとする。

手順

– ロギングのタイプを循環ロギングからアーカイブ・ロギングに変更する。
– データベースのオフライン・バックアップを取得する。
– バックアップ後の全てのログを取得する。
– 取得したバックアップイメージを用いて、リストアする。
– 取得したログを用いて、ロールフォワードを実行し、データベースをリカバリーする。

1-2. リカバリーのシナリオ (1/2)



1-2. リカバリーのシナリオ (2/2)

	シナリオ	時刻
①	データベースのオフライン・バックアップを取得する。	T0
②	データベースに対する更新として、テーブル TAB1へデータを INSERT する。	T1
③	ログをアーカイブする。	T2
④	データベースへの障害として、DROP DATABASE を行う。	T3
⑤	①で取得したデータベースのオフライン・バックアップをリストアする。	T4
⑥	③で取得したアーカイブ・ログを使用してロールフォワードを行い、完了する。	T4～T5
⑦	データベースはログに書き出されている最後のトランザクションがコミットされた状態にリカバリーされる。	T5

1-2. 手順 (1/3)

- データベースのオフライン・バックアップを取得する。（①）

```
$ db2 backup db SAMPLE to /work
バックアップは成功しました。このバックアップ・イメージのタイム・スタンプは  20250807101756 です。
```

- データベースに接続、更新を行う。（②）

```
$ db2 "insert into TAB1 values (1,'TEST’)”
DB20000I  SQL コマンドが正常に完了しました。
```

- ログをアーカイブする。（③）

```
$ db2 terminate
DB20000I  TERMINATE コマンドが正常に完了しました。
$ db2 archive log for db SAMPLE
DB20000I  ARCHIVE LOG コマンドが正常に完了しました。
```

- データベースへの障害として、DROP DATABASE を行う。（④）

```
$ db2 drop db SAMPLE
DB20000I  DROP DATABASE コマンドが正常に完了しました。
```

1-2. 手順 (2/3)

- ①で取得したオフライン・バックアップイメージより、リストアを行う。（⑤）

```
$ db2 restore db SAMPLE from /work taken at 20250807101756
DB20000I  RESTORE DATABASE コマンドが正常に完了しました。
```

- ③で取得したアーカイブ・ログを使用して、ロールフォワードを行う。（⑥）

```
$ db2 rollforward db SAMPLE to end of logs
                           ロールフォワード状況
 入力データベース別名                    = SAMPLE
 状況を返したメンバーの数                = 1
 メンバー ID                             = 0
 ロールフォワード状況                    = DB  作業中
 次に読み込むログ・ファイル              = S0000002.LOG
 処理したログ・ファイル                  = S0000000.LOG - S0000000.LOG
 最後にコミットしたトランザクション      = 2025-08-07-01.17.57.000000 UTC
DB20000I  ROLLFORWARD コマンドが正常に完了しました。
```

- ロールフォワードを完了させる。

```
$ db2 rollforward db SAMPLE complete
     (省略)
 ロールフォワード状況                    = 非ペンディング
 次に読み込むログ・ファイル              =
 処理したログ・ファイル                  = S0000000.LOG - S0000001.LOG
 最後にコミットしたトランザクション      = 2025-08-07-01.17.57.000000 UTC
DB20000I  ROLLFORWARD コマンドが正常に完了しました。
```

1-2. 手順 (3/3)

- データベースへ接続し、オフライン・バックアップ後に実施した更新が反映され、このT5時点で過去のT2の状態にリカバリーされていることを確認する。（⑦）

```
$ db2 "select * from TAB1"

C1          C2
----------- --------------------
          1 TEST

  1 レコードが選択されました。
```

1-3. オフライン・バックアップ取得時点以降の任意の時点にリカバリーする

内容

– オフライン・バックアップを使用したリカバリーの手順

概要

– データデータベースのオフライン・バックアップと、バックアップ取得後のアーカイブ・ログを使用して、特定時刻までデータベースをリカバリーする手順を紹介する。

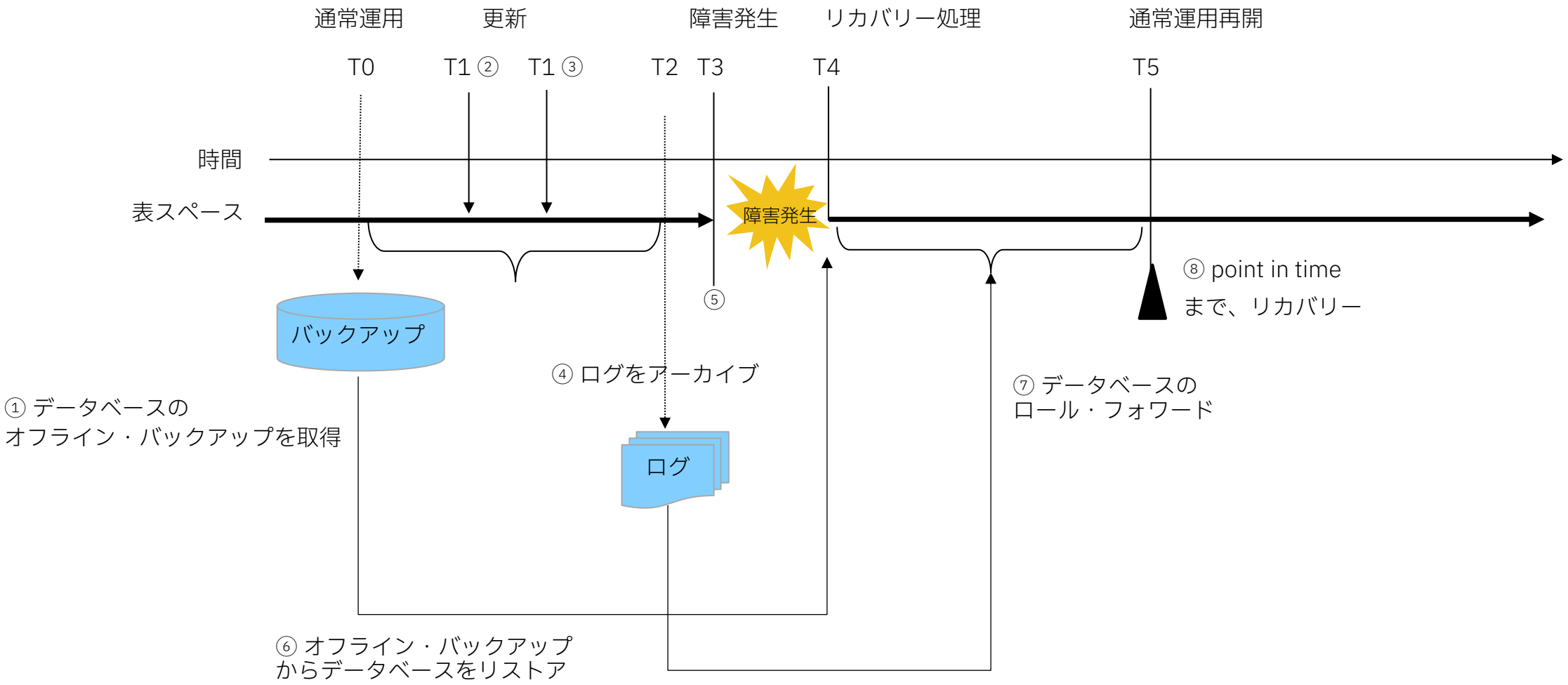
前提

– ロールフォワード・リカバリーを利用するための、ロギングのタイプをアーカイブ・ロギングとする。

手順

– データデータベースのオフライン・バックアップを取得する。
– データベースに対する更新処理として、テーブル TAB1 へデータを INSERT する。
– 取得したバックアップイメージを用いて、リストアする。
– ロールフォワードを実行し、特定時刻の処理まで、データベースをリカバリーする。

1-3. リカバリーのシナリオ (1/2)



1-3. リカバリーのシナリオ (2/2)

	シナリオ	時刻
①	データベースのオフライン・バックアップを取得する。	T0
②	データベースへの更新として、テーブル TAB1へデータを INSERT する。 日付と時刻を確認する。	T1
③	データベースへの更新として、テーブル TAB1へデータを INSERT する。 日付と時刻を確認する。	T1
④	ログをアーカイブする。	T2
⑤	データベースへの障害として、DROP DATABASE を行う。	T3
⑥	①で取得したデータベースのオフライン・バックアップをリストアする。	T4
⑦	④で取得したアーカイブ・ログを使用して②で確認した時刻までロールフォワードを行い、完了する。	T4～T5
⑧	データベースは、②のトランザクションがコミットされた状態にリカバリーされる。	T5

1-3. 手順 (1/3)

- データベースのオフライン・バックアップを取得する。（①）

```
$ db2 backup db SAMPLE to /work
バックアップは成功しました。このバックアップ・イメージのタイム・スタンプは  20250807102957 です。
```

- データベースへの更新処理として、テーブル TAB1にデータを INSERT し、日付と時刻を確認する。（②）

```
$ db2 "insert into TAB1 values ( 1, 'TEST’ )”
DB20000I  SQL コマンドが正常に完了しました。
$ date
2025年  8月  7日 木曜日 10:31:22 JST
```

- データベースへの更新処理として、テーブル TAB1 にデータを INSERT し、日付と時刻を確認する。（③）

```
$ db2 "insert into TAB1 values ( 2, 'TEST' )"
DB20000I  SQL コマンドが正常に完了しました。
$ date
2025年  8月  7日 木曜日 10:32:03 JST
```

- ログをアーカイブする。（④）

```
$ db2 archive log for db SAMPLE
20000I  ARCHIVE LOG コマンドが正常に完了しました。
```

1-3. 手順 (2/3)

- データベースの障害として、DROP DATABASE を行う。（⑤）

```
$ db2 drop db SAMPLE
20000I  DROP DATABASE コマンドが正常に完了しました。
```

- ①で取得したオフライン・バックアップイメージより、リストアを行う。（⑥）

```
$ db2 restore db SAMPLE from /work taken at 20250807102957
DB20000I  RESTORE DATABASE コマンドが正常に完了しました。
```

- ④で取得したアーカイブ・ログを使用して、②の時点までロールフォワードを行い、停止する。（⑦）

```
$ db2 rollforward db SAMPLE to 2025-08-07-10.31.22 using local time and stop
                                  ロールフォワード状況
 入力データベース別名                    = SAMPLE
 状況を返したメンバーの数                = 1
 メンバー ID                            = 0
 ロールフォワード状況                    = 非ペンディング
 次に読み込むログ・ファイル              =
 処理したログ・ファイル                  = S0000002.LOG - S0000003.LOG
 最後にコミットしたトランザクション      = 2025-08-07-10.29.57.000000 Local
DB20000I  ROLLFORWARD コマンドが正常に完了しました。
```

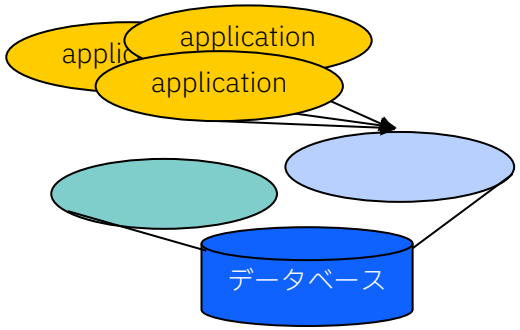
1-3. 手順 (3/3)

- データベースに接続し、T1②の時点までリカバリーされていることを確認する。（⑧）

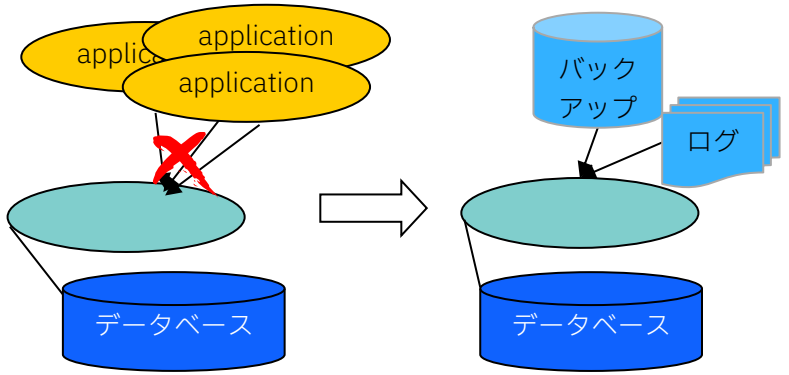
```
$ db2 "select * from TAB1"
C1          C2
----------- --------------------
          1 TEST

  1 レコードが選択されました。
```

参考：リストア操作後の接続要求のブロック

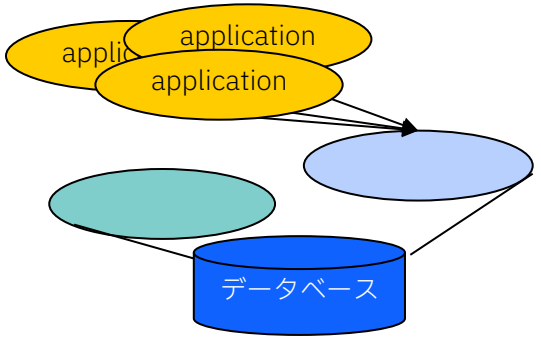
- オフライン・バックアップイメージからリストア操作を行うと、リストア操作完了時点でデータベースへの接続が可能な状態となる。その場合、当該データベースのログ・チェーンが変更され、オフライン・バックアップ後に取得したログを利用してのロールフォワード操作が行えなくなる。
- ACTIVATE DATABASE コマンドを利用した際も、ログ・チェーンが変更され、同様にロールフォワード操作が行えなくなる。
- 従って、オフライン・バックアップ後に取得したログを利用してのロール・フォワード操作を必要とする場合は、ACTIVATE DATABASE 操作を行わないことと、データベースへの接続要求をブロックすることの、の2点が必要となる。
- データベースへの接続要求をブロックする運用方法としては、以下の例がある。
  - あらかじめアプリケーション接続用のデータベース別名とリストア操作用のデータベース別名を作成する。
  - リストアを行う場合は、事前にアプリケーション接続用のデータベース別名を UNCATALOG する。
  - リストア操作は、リストア操作用データベース別名に対して行う。
  - ロールフォワード操作に関しても、リストア操作用データベース別名に対して行う。
  - ロールフォワード操作が完了し、データベースが接続可能な状況になった時点で、アプリケーション接続用のデータベース別名を CATALOG し、アプリケーションから接続可能とする。



アプリケーションは、アプリケーション接続用データベース別名を利用して接続を行う



アプリケーション接続用別名を UNCATALOG し、アプリケーションからの接続が不可となるようにした上で、リストア操作を開始する



ロール・フォワードが完了後にデータベース接続用別名を CATALOG する

オンライン・バックアップを使用したリカバリー

02

2. オンライン・バックアップを使用したリカバリー

2-1. オンライン・バックアップ取得時点から最後のログまで、または時刻指定でリカバリーする

2-2. INCLUDE LOGSオプションを使用して取得したオンライン・バックアップの時点から、
最後のログまでリカバリーする

2-1. オンライン・バックアップ取得時点から最後のログまで、または時刻指定でリカバリーする

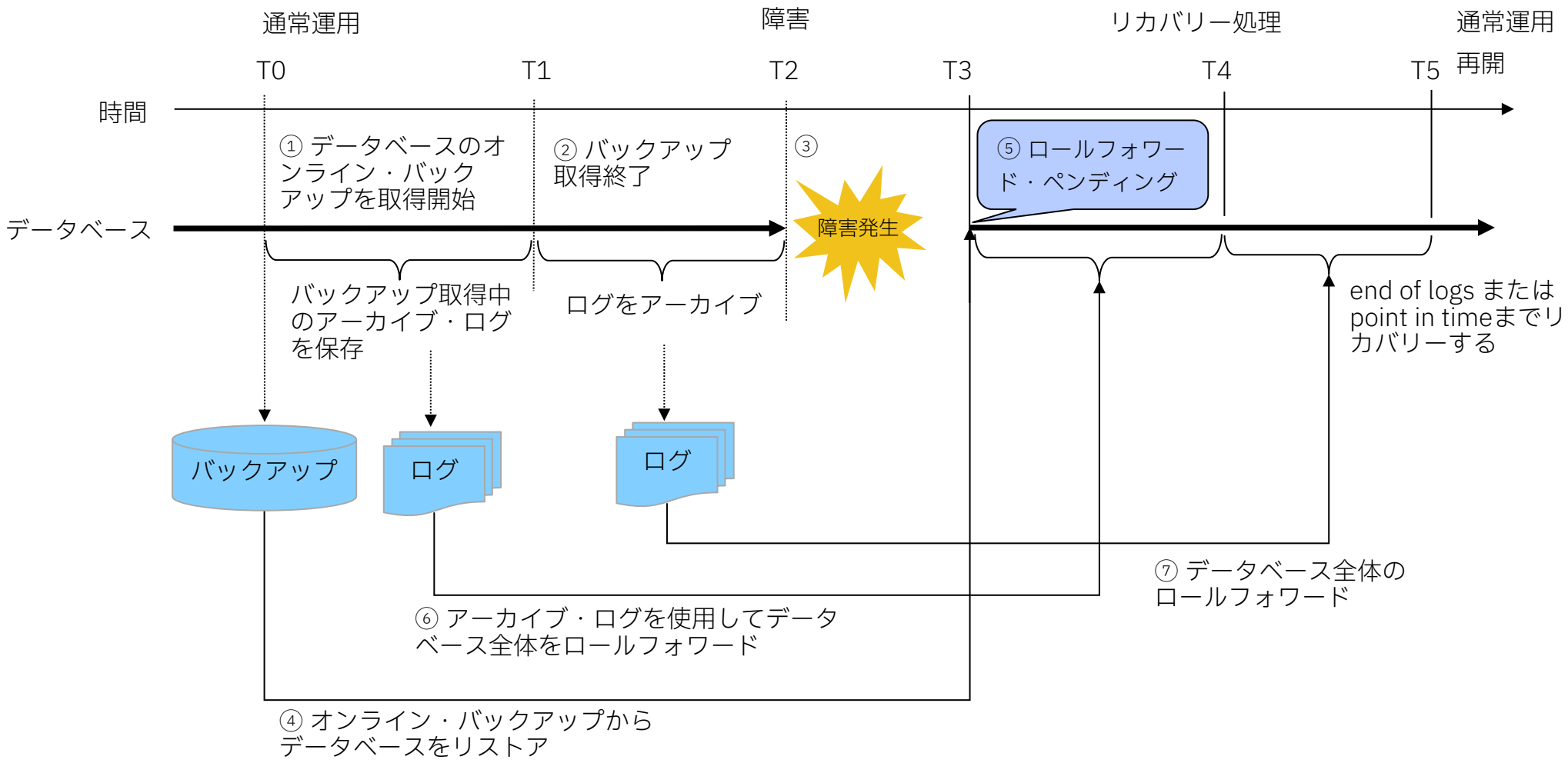
内容
– オンライン・バックアップを使用したリカバリーの手順

概要
– 「オンライン」とは、バックアップ操作中に他のアプリケーションが、そのデータベースに接続できるという意味である。
– オフライン・バックアップより時間がかかる。
– BACKUP コマンドで ONLINE オプションを指定し、実行することで、オンライン・バックアップが取得できる。

前提
– データベースが「リカバリー可能」であることが必要である。
– オンライン・バックアップの取得、およびオンライン・バックアップ開始以降のログの補完が必要となる。

手順
– データベースのオンライン・バックアップを取得する。
– バックアップ後すべてのログを取得する。
– 取得したバックアップイメージを用いて、リストアする。
– 取得したログを用いて、ログの最後まで（end of logs）、または指定時刻（point in time）のロールフォワードを実行し、データベースをリカバリーする。

2-1. リカバリーのシナリオ (1/2)



2-1. リカバリーのシナリオ (2/2)

	シナリオ	時刻
①	オンラインでデータベースのバックアップを取得開始する。バックアップ取得中も他のアプリケーションがデータベースに対して更新操作を行う。	T0
②	オンライン・バックアップの取得が完了する。その後もアプリケーションがデータベースに対して更新操作を行う。	T1
③	データベースに障害が発生し、データベースが使用不能になったとする。	T2
④	①で取得したオンライン・バックアップイメージからデータベースをリストアする。	T3
⑤	リストア終了後、データベースはロールフォワード・ペンディング状態になる。	
⑥	オンライン・バックアップ取得中のログを使用して、データベースのロールフォワードを行う。	T3~T4
⑦	⑥に続いてバックアップ取得後のログを使用して、ログの最後まで、または時刻を指定してロールフォワードを行い、停止する。	T4~T5

2-1. 手順 (1/3)

- データベースのバックアップをオンラインで取得する。（①） オンラインでバックアップを取得しているため、バックアップ取得中もデータベースの更新が可能である。バックアップ取得終了時点のデータベースの状態をT1で表す。（②）

```
$ db2 backup db SAMPLE online to /work
バックアップは成功しました。このバックアップ・イメージのタイム・スタンプは 20250820131928 です。
```

- データベースの更新処理を行う。その後、データベース障害が発生し、データベースが使用不能になったとする。（③）
- データベースをリストアする。（④）
  - SQL2539W のメッセージは、バックアップイメージが既存のデータベースを上書きするという警告である。

```
$ db2 restore db SAMPLE from /work taken at 20250820131928 replace existing
SQL2539W  リストアするバックアップ・イメージに指定された名前がターゲット・データベースの名前と同じになっています。 バックアップ・イメージ・データベースと同じ既存データベースをリストアすると、現行のデータベースがバックアップ・バージョンによって上書きされます。
DB20000I  RESTORE DATABASE コマンドが正常に完了しました。
```

- リストアが終了すると、データベースはロールフォワード・ペンディング状態になる。（⑤）
- この状態では、データベースに接続しようとしても、SQL1117N（ロールフォワード保留中）のエラーで接続できない。

```
$ db2 connect to SAMPLE
SQL1117N  「ロールフォワード・ペンディング」のために、データベース "SAMPLE" の接続または
アクティブ化を行うことはできません。  SQLSTATE=57019
```

- なお、データベースのオンライン・バックアップからのリストアで、without rolling forward オプションを指定すると、SQL2537N（リストア後のロールフォワードが必要）となり、リストアは行われない。

2-1. 手順 (2/3)

- ロールフォワード・ペンディング中であることは、データベースの構成情報を表示させることによっても確認できる。

```
$ db2 get db cfg for SAMPLE
 データベースのデータベース構成    SAMPLE

(省略)
 バックアップ・ペンディング                                   = NO
 すべてのコミット済みトランザクションはディスクに書き込み済み  = NO
 ロールフォワード・ペンディング                               = DATABASE
 リストア・ペンディング                                       = NO
(省略)
```

2-1. 手順 (3/3)

- オンライン・バックアップをリストアした場合には、少なくともオンライン・バックアップが終了した時点までデータベースのロールフォワードを行う必要がある。この終了時点で、データベースの状態は過去のT1の状態と同じになる（⑥）
- ログの最後まで、またはオンライン・バックアップが終了した時点以降の時刻を正しく指定することにより、一度にロールフォワードを行い、データベースのロールフォワード・ペンディング状態を解除することが可能である。（⑦）

- 以下は、using local timeで時刻を指定した、point in timeリカバリーの例
  - ログの最後まで適用する場合は、end of logsを指定する

```
$ db2 rollforward db SAMPLE to 202508-08-20-13.19.30 using local time and stop
                                ロールフォワード状況

 入力データベース別名                       = SAMPLE
 状況を返したメンバーの数                   = 1

 メンバー ID                                = 0
 ロールフォワード状況                       = 非ペンディング
 次に読み込むログ・ファイル                 =
 処理したログ・ファイル                 = S0000007.LOG - S0000008.LOG
 最後にコミットしたトランザクション     = 2025-08-20-13.19.28.000000 Local

DB20000I  ROLLFORWARD コマンドが正常に完了しました。
```

2-1. 手順（参考）

- オンライン・バックアップが終了した時点以前を指定してロールフォワードを行おうとすると、SQL1275Nのエラーで失敗する。
- オンライン・バックアップイメージからリストアを行い、 end of logs やpoint in time でログを適用することなく、ロールフォワードを停止（stop）や完了（complete）した場合、SQL1276Nエラーが発生する。
- using local time オプションを指定した場合は、GMT時刻の指定ではなく、現地時刻を指定することが可能である。

```
$ db2 rollforward db SAMPLE to 202508-08-20-13.19.26 using local time and stop
SQL1275N  ノード "0" のデータベース "SAMPLE"には指定された時刻より後の情報が含まれるため、ロールフ
ォワード・ユーティリティーに渡される停止時刻は、タイム・スタンプ "2025-08-20-13.19.28.000000 Local"
以降にする必要があります。
```

- ロールフォワードが完了していない状態の場合、データベースは、ロールフォワード・ペンディング中=DATABASE、リストア・ペンディング中＝NOとなる。すなわち、このときに可能な操作は、以下のいずれかの操作のみである。
  - 指定する停止時刻をエラーメッセージに表示された時刻以降にして、ロールフォワードを再び行う。
  - 再度バックアップからのリストアを行う。

```
$ db2 get db cfg for SAMPLE
 データベースのデータベース構成    SAMPLE
(省略)
 バックアップ・ペンディング                                  = NO
 すべてのコミット済みトランザクションはディスクに書き込み済み  = NO
 ロールフォワード・ペンディング                              = DATABASE
 リストア・ペンディング                                      = NO
(省略)
```

2-2. INCLUDE LOGSオプションを使用して取得したオンライン・バックアップの時点から、最後のログまでリカバリーする

内容

– オンライン・バックアップを使用したリカバリーの手順

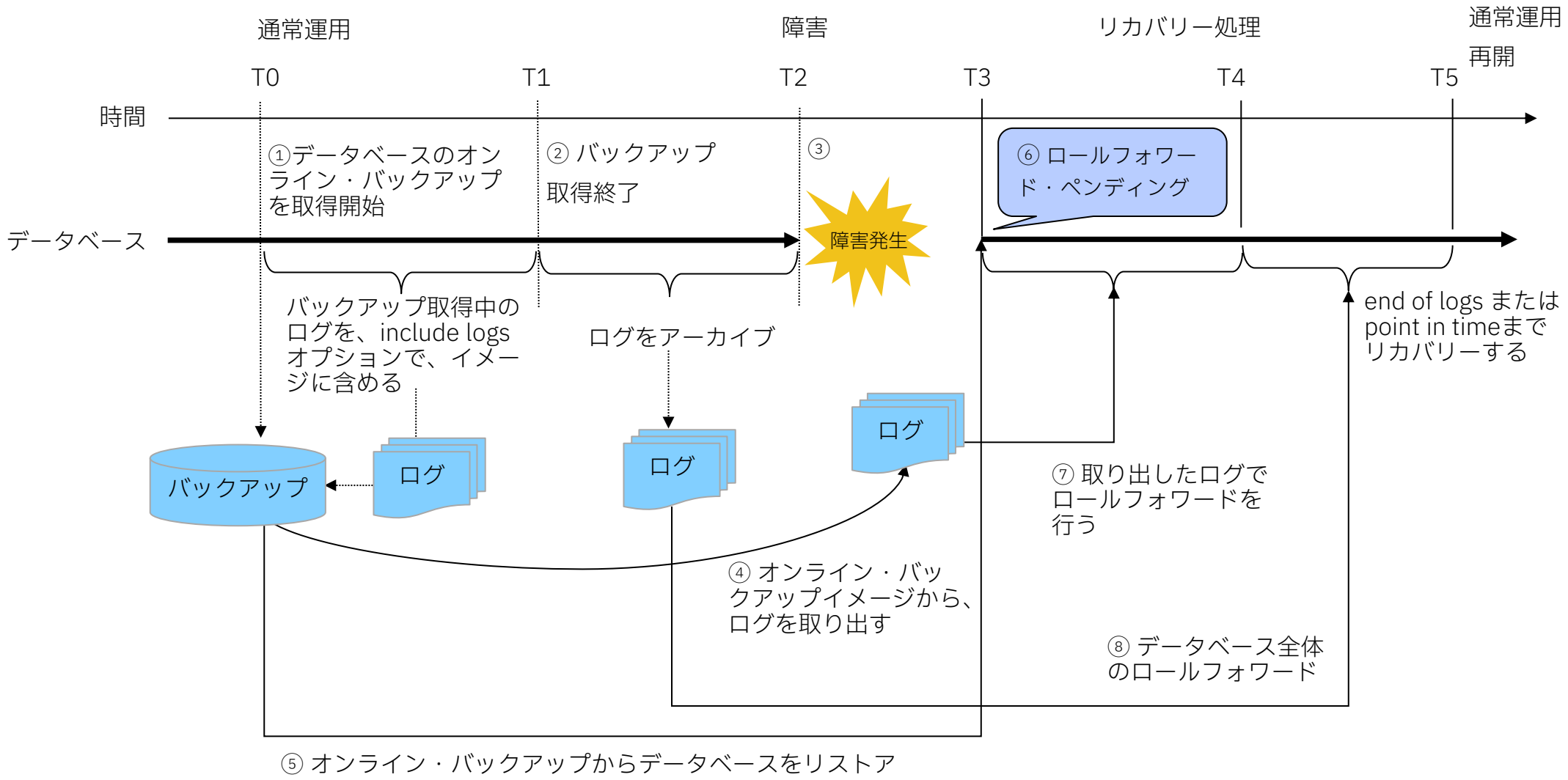
概要

– オンライン・バックアップの場合、BACKUP コマンドの INCLUDE LOGS オプションが有効である。
– INCLUDE LOGS オプションを指定することで、当該のバックアップ・イメージからリストアおよびロールフォワードを行う際に必要となるログ・ファイルを、バックアップ・イメージに含めることが可能となる。
– このオプションは、BACKUP コマンドの ONLINE オプションが指定されている場合のみ有効である。

手順

– データベースのオンライン・バックアップを INCLUDE LOGS オプションをつけて、取得する。
– 取得したバックアップイメージから、ログを取り出す。
– 取得したバックアップイメージを用いて、リストアを行う。
– 取り出したログの保管場所を指定して、ロールフォワードを実行する。

2-2. リカバリーのシナリオ (1/2)



2-2. リカバリーのシナリオ (2/2)

	シナリオ	時刻
①	オンラインでデータベースのバックアップを取得開始する。INCLUDE LOGS オプションを指定し、ログをバックアップイメージに含める。	T0
②	オンライン・バックアップの取得が完了する。 その後もアプリケーションがデータベースに対して更新操作を行う。	T1
③	データベースに障害が発生し、データベースが使用不能になったとする。	T2
④	①で取得したオンライン・バックアップイメージからログを取り出す。	
⑤	①で取得したオンライン・バックアップイメージからデータベースをリストアする。	T3
⑥	リストア終了後、データベースはロールフォワード・ペンディング状態になる。	
⑦	④で取り出したログを使用して、データベースのロールフォワードを行う。	T3~T4
⑧	ログの最後まで、または時刻を指定してロールフォワードを行い、停止する。	T4~T5

2-2. 手順 (1/4)

- データベースのバックアップをオンラインで取得する。バックアップ取得の際に INCLUDE LOGS オプションを指定し、ログをバックアップイメージに含める。（①）オンライン・バックアップの取得が完了する。（②）

```
$ db2 backup db SAMPLE online to /work include logs
バックアップは成功しました。このバックアップ・イメージのタイム・スタンプは 20250826144245 です。
```

- データベースの更新処理を行う。

```
$ db2 "insert into T1 values (1, 'TEST')"
DB20000I  SQL コマンドが正常に完了しました。
```

- その後データベースに障害が発生し、データベースが使用不能になったとする。（③）

```
$ db2 drop db SAMPLE
DB20000I  DROP DATABASE コマンドが正常に完了しました。
```

- リストアコマンドの LOGS オプションおよび LOGTARGET オプションを指定することで、バックアップイメージに含めたログを取り出す。（④）

```
$ db2 restore db SAMPLE logs from /work taken at 20250826144245 logtarget /home/db2inst1/logs
DB20000I  RESTORE DATABASE コマンドが正常に完了しました。
$ ls -l /home/db2inst1/logs/NODE0000/LOGSTREAM0000/
合計 12
-rw-------. 1 db2inst1 db2iadm1 12288 8月 26 14:52 S0000004.LOG
```

- オンライン・バックアップイメージからデータベースをリストアする。（⑤）

```
$ db2 restore db SAMPLE from /work taken at 20250826144245
DB20000I  RESTORE DATABASE コマンドが正常に完了しました。
```

2-2. 手順 (2/4)

- リストアされたデータベースはロールフォワード・ペンディングとなっている。（⑥）

```
$ db2 get db cfg for SAMPLE
        データベースのデータベース構成    SAMPLE
  (省略)
 バックアップ・ペンディング                                    = NO
 すべてのコミット済みトランザクションはディスクに書き込み済み  = NO
 ロールフォワード・ペンディング                               = DATABASE
 リストア・ペンディング                                        = NO
  (省略)
```

- ④で取り出したログで、ロールフォワードを行う。（⑦）
  - OVERFLOW LOG PATH として LOGTARGET オプションで指定したディレクトリーを指すことで、オンライン・バックアップイメージからのリストアに必要となるログを適用している。

```
$ db2 "rollforward db SAMPLE to end of logs overflow log path (/home/db2inst1/logs)”
                                ロールフォワード状況
 入力データベース別名                    = SAMPLE
 状況を返したメンバーの数                = 1
 メンバー ID                             = 0
 ロールフォワード状況                    = DB  作業中
 次に読み込むログ・ファイル              = S0000006.LOG
 処理したログ・ファイル                  = S0000004.LOG - S0000004.LOG
 最後にコミットしたトランザクション      = 2025-08-26-14.42.45.000000 Local
DB20000I  ROLLFORWARD コマンドが正常に完了しました。
```

2-2. 手順 (3/4)

- ログの最後までロールフォワードを行い、停止する。（⑧）

```
$ db2 rollforward db SAMPLE to end of logs and complete
                                 ロールフォワード状況
 入力データベース別名                       = SAMPLE
 状況を返したメンバーの数                   = 1
 メンバー ID                                = 0
 ロールフォワード状況                       = 非ペンディング
 次に読み込むログ・ファイル                 =
 処理したログ・ファイル                 = S0000004.LOG - S0000005.LOG
 最後にコミットしたトランザクション     = 2025-08-26-14.42.45.000000 Local
DB20000I  ROLLFORWARD コマンドが正常に完了しました。

$ db2 get db cfg for SAMPLE
       データベースのデータベース構成    SAMPLE
  (省略)
 バックアップ・ペンディング                                  = NO
 すべてのコミット済みトランザクションはディスクに書き込み済み  = YES
 ロールフォワード・ペンディング                              = NO
 リストア・ペンディング                                      = NO
  (省略)
```

2-2. 手順 (4/4)

- データベースに接続し、更新内容を確認する。

```
$ db2 "select * from T1"

C1          C2
----------- ----------
          1 TEST

  1 レコードが選択されました。
```

2-2. 手順（参考）

- 手順④、⑤のように、個別にログを取り出すのではなく、リストア実行時に LOGTARGET オプションを指定することで、必要となるログを取り出すことも出来る。

```
$ db2 restore db SAMPLE from /work taken at 20250826144245 logtarget /home/db2inst1/logs
DB20000I  RESTORE DATABASE コマンドが正常に完了しました。
```

表スペース・バックアップを使用したリカバリー

03

3. 表スペース・バックアップを使用したリカバリー

内容
表スペース・バックアップを使用したリカバリーの手順

3-1. 表スペース・バックアップをリストアし、最後のログまでリカバリーする

3-2. 表スペース・バックアップをリストアし、特定時間までリカバリーする

3-1. 表スペース・バックアップをリストアし、最後のログまでリカバリーする

概要

– 特定のユーザー表スペースのオフライン・バックアップを使用して、表スペースをリストアし、破損した表スペースのリカバリーを、最後のログまで行う

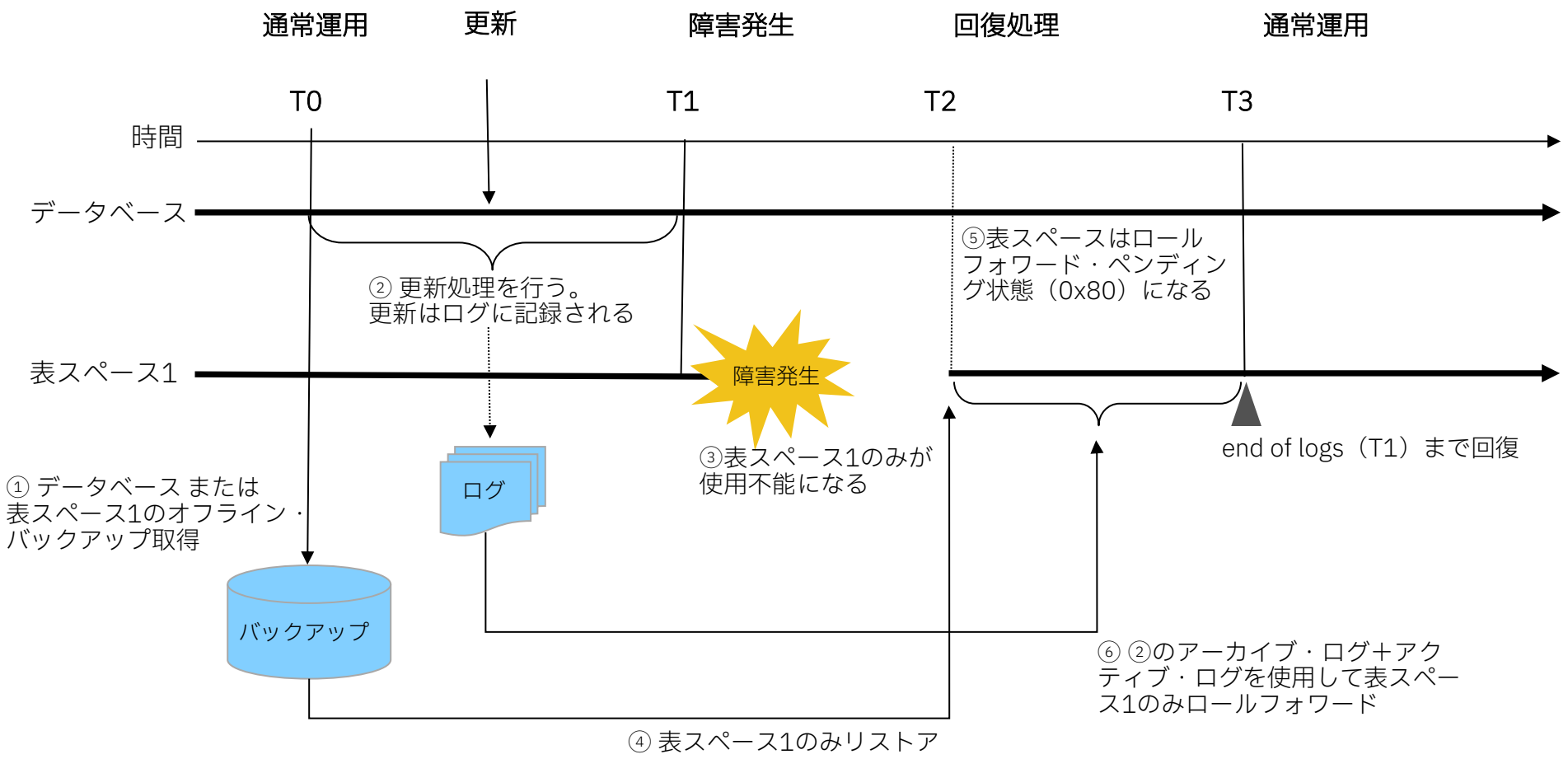
前提

– 表スペースのリストアのソースとなるバックアップ・イメージ
  • オフラインバックアップ：データベース全体、または回復対象の表スペースを含む表スペース・バックアップ
  • オンラインバックアップ：データベース全体、または回復対象の表スペースを含む表スペース・バックアップ（※この場合、バックアップ取得以降～回復時点までのトランザクション・ログが必要）
– 特定の表スペースのリカバリーのみでデータの整合性が保たれる場合に選択可能なリカバリー方法であることに留意する
– バックアップ取得後の表の削除や列追加等、システムカタログ表の更新を伴う操作については特定の表スペース・リカバリーでは回復不可能(※システムカタログ表もリカバリーする必要あり)
– 次ページ以降では、メディア障害で表スペース TBS1 のみ使用不能になった場合の復旧手順として、バックアップ・イメージから TBS1 をリストアし、END OF LOGS までロールフォワードして整合性を回復する手順を示す

手順

– 対象となる表スペースのバックアップを取得する
– 取得した表スペースのバックアップより、リストアを行う
– リストアした表スペースをロールフォワードする

3-1. リカバリーのシナリオ (1/2)



3-1. リカバリーのシナリオ (2/2)

	シナリオ	時刻
①	オフラインでデータベースまたは表スペースのバックアップを取得する。	T0
②	データベースの更新操作を行う。更新はログに記録される。	
③	データベースに障害が発生し、1つの表スペースのみが使用不能になったとする。	T1
④	①で取得したデータベースまたは表スペースのバックアップより表スペースのみをリストアする。	T2
⑤	表スペースはロールフォワード・ペンディング状態になる。	
⑥	②のログを使用して、表スペースをログの最後までロールフォワードする。	T3

3-1. 手順 (1/4)

- 表スペース（TBS1）のオフライン・バックアップを取得する。この時点のデータベースをT0時点で表する。（①）

```
$ db2 "backup database SAMPLE tablespace (TBS1) to /work"
バックアップは成功しました。このバックアップ・イメージのタイム・スタンプは20250730144612 です。
```

- データベースの更新処理を行う。（②）
- データベースに障害が発生し、表スペース TBS1 のみ使用不能になったとする。（③）
- 表スペース TBS1 のみリストアする。（④）

```
$ db2 "insert into T1 values (2, 'TEST')"
DB20000I  SQL コマンドが正常に完了しました。
$ db2 deactivate database SAMPLE
DB20000I  DEACTIVATE DATABASE コマンドが正常に完了しました。
$ mv /home/db2inst1/tbs1  /work/tbs1.01
$ db2 "select * from T1"

C1          C2
----------- --------------------
SQL0290N  表スペース・アクセスが許されていません。SQLSTATE=55039
$ db2 "restore database SAMPLE tablespace (TBS1) from /work taken at 20250730144612"
DB20000I  RESTORE DATABASE コマンドが正常に完了しました。
```

3-2. 手順 (2/4)

- リストアが終了すると、表スペース TBS1 はオフライン・バックアップを取得した時点（T0時点）と同じになる。ただし、この表スペース TBS1 はロールフォワード・ペンディング状態になる。
- この状態では、データベースに接続は可能であるが、ロールフォワード・ペンディング状態の表スペースへのアクセスは、SQL0290N でエラーとなる。例えば、表スペース TBS1 上に作成したテーブル T1 に対して、SELECT 文を実行すると、次のようになる。

```
$ db2 "select * from T1"
C1          C2
----------- --------------------
SQL0290N  表スペース・アクセスが許されていません。SQLSTATE=55039
```

- 表スペース TBS1 がロールフォワード・ペンディング中であることは、データベースの構成情報を表示させることによっても確認できる。

```
$ db2 get db cfg for SAMPLE
     データベースのデータベース構成    SAMPLE
     （省略）
 バックアップ・ペンディング                                      = NO

 すべてのコミット済みトランザクションはディスクに書き込み済み  = YES
 ロールフォワード・ペンディング                                  = TABLESPACE
 リストア・ペンディング                                          = NO
     （省略）
```

3-1. 手順 (3/4)

- LIST TABLESPACES コマンドで、どの表スペースがロールフォワード・ペンディング中であるかを確認することが出来る。

```
$ db2 list tablespaces
       現在のデータベースの表スペース
    （省略）
表スペース ID                        = 3
 名前                                = TBS1
 タイプ                              = データベース管理スペース
 内容                                = すべての永続データ。 LARGE 表スペース。
 状態                                = 0x0080
   詳しい説明:
     ロールフォワード・ペンディング
```

- ロールフォワード・ペンディング状態を解除するために、表スペース TBS1 のロールフォワードを、時刻指定 （POINT IN TIME） あるいはログの最後まで （END OF LOGS） にて行、ロールフォワードを停止させる必要がある。（⑥）

```
$ db2 "rollforward database SAMPLE to end of logs and stop tablespace (TBS1) "
                                ロールフォワード状況

 入力データベース別名                = SAMPLE
 状況を返したメンバーの数            = 1

 メンバー ID                         = 0
 ロールフォワード状況                = 非ペンディング
 次に読み込むログ・ファイル          =
 処理したログ・ファイル              =  -
 最後にコミットしたトランザクション  = 2025-07-30-05.46.40.000000 UTC
DB20000I  ROLLFORWARD コマンドが正常に完了しました。
```

3-1. 手順 (4/4)

- ロールフォワード・ペンディング状態の解除は、データベースの構成情報より確認できる。

```
$ db2 get db cfg for SAMPLE
     データベースのデータベース構成     SAMPLE
     (省略)
 バックアップ・ペンディング                                      = NO

 すべてのコミット済みトランザクションはディスクに書き込み済み  = YES
 ロールフォワード・ペンディング                                  = NO
 リストア・ペンディング                                          = NO
     (省略)
```

- また、LIST TABLESPACES コマンドでも、ロールフォワード・ペンディング状態が解除されたことを確認することが出来る。

```
$ db2 list tablespaces
       現在のデータベースの表スペース
     (省略)
表スペース ID                              = 3
 名前                                      = TBS1
 タイプ                                    = データベース管理スペース
 内容                                      = すべての永続データ。 LARGE 表スペース。
 状態                                      = 0x0000
   詳しい説明:
      正常
```

3-2. 表スペース・バックアップをリストアし、特定時間までリカバリーする

概要

– 特定のユーザー表スペースのオフライン・バックアップを使用して、表スペースをリストアし、破損した表スペースのリカバリーを、特定時間（ POINT IN TIME ）まで行う

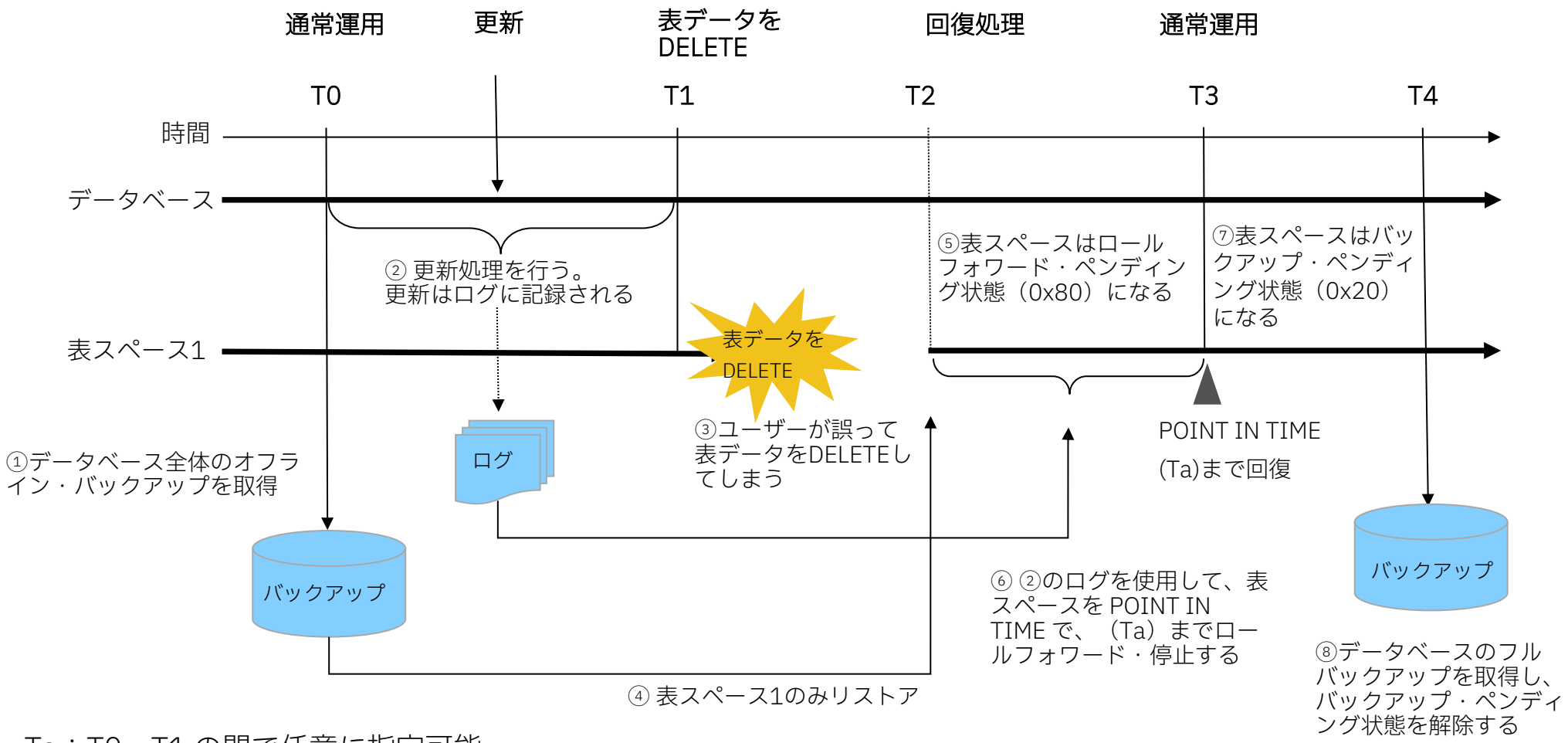
前提

– 表スペースのリストアのソースとなるバックアップ・イメージ
  • オフラインバックアップ：データベース全体、または回復対象の表スペースを含む表スペース・バックアップ
  • オンラインバックアップ：データベース全体、または回復対象の表スペースを含む表スペース・バックアップ（※この場合、バックアップ取得以降～回復時点までのトランザクション・ログが必要）
– 特定の表スペースのリカバリーのみでデータの整合性が保たれる場合に選択可能なリカバリー方法であることに留意する
– バックアップ取得後の表の削除や列追加等、システムカタログ表の更新を伴う操作については表スペース・リカバリーでは回復不可能(※システムカタログ表もリカバリーする必要あり)
– 次ページ以降では、表のデータを誤ってDELETEしてしまったのリカバリー手順として、その表が格納される表スペースをバックアップ・イメージからリストアすることによって、表データをDELETE前の状態に戻す手順を示す

手順

– 対象となる表スペースのバックアップを取得する
– 取得した表スペースのバックアップより、リストアを行う
– リストアした表スペースをPOINT IN TIME でロールフォワードする
– DBのフルバックアップを取得する

3-2. リカバリーのシナリオ (1/2)



Ta：T0～T1 の間で任意に指定可能

3-2. リカバリーのシナリオ (2/2)

	シナリオ	時刻
①	オフラインでデータベースのバックアップを取得する。	T0
②	データベースの更新操作を行う。更新はログに記録される。	
③	ユーザーが誤って表データをDELETEしてしまったとする。	T1
④	①で取得したデータベースのバックアップより、表スペースのみをリストアする。	T2
⑤	表スペースはロールフォワード・ペンディング状態になる。	
⑥	②のログを使用して、表スペースを POINT IN TIME で、（Ta）までロールフォワードを停止する。	T3
⑦	表スペースはバックアップ・ペンディング状態になる。	
⑧	データベースのフルバックアップを取得し、バックアップ・ペンディング状態を解除する。（それでも時間がない人は表スペースバックアップを取ってデータベースの利用を再開することも出来る）	T4

3-2. 手順 (1/8)

- データベースのオフライン・バックアップを取得する。この時点のデータベースをT0時点で表す。（①）

```
$ db2 backup db SAMPLE to /work
バックアップは成功しました。このバックアップ・イメージのタイム・スタンプは 20251010173127 です。
```

- データベースの更新処理を行う。（②）

```
$ db2 "insert into T1 values (3,'TEST')"
DB20000I  SQL コマンドが正常に完了しました。

$ db2 "select * from T1"

C1          C2
----------- --------------------
          1 Backup
          2 Recovery
          3 TEST

  3 レコードが選択されました。
```

3-2. 手順 (2/8)

- ユーザーが誤って表データをDELETEしてしまったと仮定し（③）、Point in Timeリカバリーで適切な時刻を指定できるよう、DELETE の実行時刻を確認・記録する。

```
$ db2 "VALUES CURRENT TIMESTAMP"
db2 "DELETE FROM T1 WHERE C1=3"
db2 "COMMIT"

1
--------------------------
2025-10-10-17.32.55.130048
  1 レコードが選択されました。

DB20000I  SQL コマンドが正常に完了しました。
DB20000I  SQL コマンドが正常に完了しました
```

- 表スペース TBS1 のみ復元する。（④）

```
$ db2 "restore database SAMPLE tablespace (TBS1) from /work taken at 20251010173127 replace
existing"
DB20000I  RESTORE DATABASE コマンドが正常に完了しました。
```

- リストアが終了すると、表スペースはオフライン・バックアップを取得した時点（T0時点）と同じになる。ただし、この表スペース TBS1 はロールフォワード・ペンディング状態になる。（⑤）

3-2. 手順 (3/8)

- この状態では、データベースは接続は可能であるが、ロールフォワード・ペンディング状態の表スペースへのアクセスは、SQL0290 でエラーとなる。例えば、表スペース TBS1 上に作成した T1 表に対して、SELECT 文を実行すると、次のようになる。

```
$ db2 "select * from T1"

C1          C2
----------- --------------------
SQL0290N  表スペース・アクセスが許されていません。
SQLSTATE=55039
```

- 表スペース TBS1 がロールフォワード・ペンディング中であることは、データベースの構成情報を表示させることによって確認できる。

```
$ db2 get db cfg for SAMPLE
     データベースのデータベース構成    SAMPLE
     (省略)
 バックアップ・ペンディング                                      = NO

 すべてのコミット済みトランザクションはディスクに書き込み済み  = YES
 ロールフォワード・ペンディング                                  = TABLESPACE
 リストア・ペンディング                                          = NO
     (省略)
```

3-2. 手順 (4/8)

- LIST TABLESPACES コマンドで、ロールフォワード・ペンディング中の表スペース情報を確認することができる。

```
$ db2 list tablespaces
        現在のデータベースの表スペース
    （省略）
表スペース ID                                = 3
 名前                                        = TBS1
 タイプ                                      = データベース管理スペース
 内容                                        = すべての永続データ。 LARGE 表スペース。
 状態                                        = 0x0080
   詳しい説明:
      ロールフォワード・ペンディング
```

3-2. 手順 (5/8)

- ロールフォワード・ペンディング状態を解除するために、表スペース TBS1 のロールフォワードを、時刻指定（POINT IN TIME）またはログの最後（END OF LOGS）まで行い、かつ停止する必要がある。Point In Time リカバリでは「最小リカバリ時間」が存在し、それより前の時刻を指定するとエラーになる。（詳細はSIL「Db2 バックアップ・リカバリー」＞「表スペース・リカバリーの考慮点（3/5） 」を参照）。MON_GET_TABLESPACE 表関数を使用して、表スペースをロールフォワードできる最も早いPOINT IN TIMEである最小リカバリー時間を判別する。

```
$ db2 "select tbsp_name, tablespace_min_recovery_time
     from table(mon_get_tablespace('TBS1', -1))"

TBSP_NAME                                        TABLESPACE_MIN_RECOVERY_TIME
------------------------------------------------------------------------------
TBS1                                              2025-10-10-17.30.13.000000

1 レコードが選択されました。
```

※今回のケースでは DDL を実行していないため、tablespace_min_recovery_time はバックアップ時刻より前となっている。そのため、実際にロールフォワードで指定できる最小時刻はバックアップ時刻であり、特別な考慮は不要だった。一方、DROP TABLE など DDL を伴う場合は、必ず最小リカバリー時間以降までロールフォワードする必要がある。

3-2. 手順 (6/8)

- ここでは、Ta 時刻 を指定しての point in time にてロールフォワードし、停止する。 DELETE 前の状態に戻すためには、T1時点（③）より前の時刻（例：2025-10-10-17.32.11）を指定する必要がある。

```
$ db2 "rollforward db SAMPLE to 2025-10-10.17.32.11 using local time and stop tablespace (TBS1)"
                                  ロールフォワード状況

 入力データベース別名                       = SAMPLE
 状況を返したメンバーの数                   = 1

 メンバー ID                                = 0
 ロールフォワード状況                       = 非ペンディング
 次に読み込むログ・ファイル                 =
 処理したログ・ファイル                     =  -
 最後にコミットしたトランザクション         = 2025-10-10-17.32.08.000000 Local
DB20000I  ROLLFORWARD コマンドが正常に完了しました。
```

- ロールフォワードを停止後、表スペース TBS1 は時刻 Ta の状態（ Ta時点）となるが、他の表スペースは時刻T2の状態である。このとき表スペース間の整合性はユーザーの責任で保つ必要がある。
- 表スペースを POINT IN TIME を指定してロールフォワードを実行・停止すると、表スペースはバックアップ・ペンディング状態となる。（⑦）

3-2. 手順 (7/8)

- 表スペース TBS1 がバックアップ・ペンディング状態であることは、LIST TABLESPACES コマンドで確認できる。

```
$ db2 list tablespaces
       現在のデータベースの表スペース
    （省略）
表スペース ID                              = 3
 名前                                      = TBS1
 タイプ                                    = データベース管理スペース
 内容                                      = すべての永続データ。 LARGE 表スペース。
 状態                                      = 0x0020
   詳しい説明:
     バックアップ・ペンディング
```

- バックアップ・ペンディング状態では、その表スペース上の表の照会は可能であるが、更新はできない。表 T1 において DELETE 操作によって削除されたデータがロールフォワードによって復元されていることを確認する。

```
$ db2 "select * from T1"
C1          C2
----------- --------------------
          1 Backup
          2 Recovery
          3 TEST
  3 レコードが選択されました。
$ db2 "insert into T1 values (4,'Insert')"
DB21034E  コマンドが、有効なコマンド行プロセッサー・コマンドでないため、 SQLステートメントとして処理されました。
SQL処理中に、次のエラーが返されました。
SQL0290N  表スペース・アクセスが許されていません。SQLSTATE=55039
```

3-2. 手順 (8/8)

- 表スペース TBS1 がバックアップ・ペンディング状態になる理由は、以下の通りである。
  - この後に時刻 Ta から 時刻 T1 のログを使用してデータベースのロールフォワードを行った場合に、表スペース TBS1 にもそのログファイルに記録されているトランザクションが適用されてしまい、表スペースの内容が正しくない状態になってしまう。
  - しかし、この表スペース TBS1 のバックアップがあれば、それをリストアすることにより、表スペース TBS1 の状態を時刻 Ta の状態にすることができる。このバックアップを必ず取得するために、表スペースはバックアップ・ペンディング状態となる。
  - 次ページ以降の手順は、この内容に関するシナリオとなる。
- データベースのフルバックアップを取得し、表スペース TBS1 のバックアップ・ペンディング状態を解除する。（⑧）

```
$ db2 "backup db SAMPLE to /work"
バックアップは成功しました。このバックアップ・イメージのタイム・スタンプは 20251010174039 です。


$ db2 list tablespaces
        現在のデータベースの表スペース
    （省略）
表スペース ID                                  = 3
 名前                                          = TBS1
 タイプ                                        = データベース管理スペース
 内容                                          = すべての永続データ。 LARGE 表スペース。
 状態                                          = 0x0000
   詳しい説明:
      正常
```

増分バックアップを使用したリカバリー

04

4. オンライン増分バックアップからデータベース全体をリカバリーする（自動リストア）

内容
– 増分バックアップを使用したリカバリーの手順

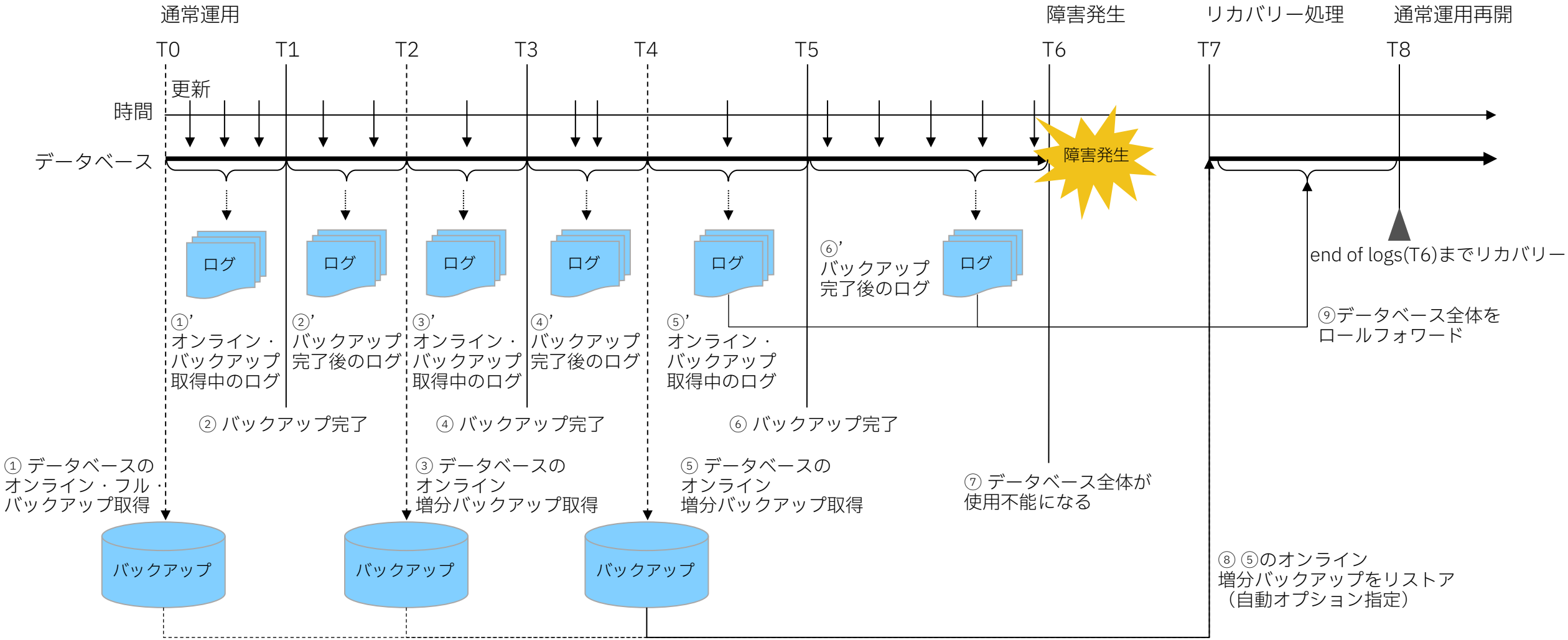
概要
– データベースのオンライン・フル・バックアップとオンライン増分バックアップ（デルタ）を使用してリストアし（自動リストア）、データベース全体をリカバリーする。

手順
– データベースのオンライン・フル・バックアップを取得する。
– データベースのオンライン増分バックアップを取得する。（データベース障害発生）
– データベースのオンライン増分バックアップを指定し、自動リストアする。
– データベースのロールフォワードを実行する。

※INCREMENTAL AUTOMATICオプションを指定した自動リストアのため、増分バックアップのリストアを順番に手動で行う必要はない。

4. リカバリーのシナリオ (1/2)

最後のオンライン・バックアップ取得完了時点までのログが最低限必要



4. リカバリーのシナリオ (2/2)

	シナリオ	時刻
①	データベースのオンライン・フル・バックアップを取得開始する。	T0
①’	オンラインバックアップ取得中の更新はログに記録される。	
②	データベースのオンライン・フル・バックアップが完了する。	T1
②’	データベースに対して更新を行い、生成されたログは保管しておく。	
③	データベースのオンライン増分バックアップ（デルタ）を取得開始する。	T2
③’	オンラインバックアップ取得中の更新はログに記録される。	
④	データベースのオンライン増分バックアップ（デルタ）が完了する。	T3
④’	データベースに対して更新を行い、生成されたログは保管しておく。	
⑤	データベースのオンライン増分バックアップ（デルタ）を取得開始する。	T4
⑤’	オンラインバックアップ取得中の更新はログに記録される。	
⑥	データベースのオンライン増分バックアップ（デルタ）が完了する。	T5
⑥’	データベースに対して更新を行い、生成されたログは保管しておく。	
⑦	データベースに障害が発生し、データベース全体が使用不能になったことを想定。	T6
⑧	⑤で取得したデータベースのオンライン増分バックアップ（ターゲットとなるバックアップ・イメージ）を、INCREMENTAL AUTOMATICオプションを指定してリストアする。（自動リストア）	T7
⑨	⑤’⑥’で保管されていたアーカイブ・ログおよびアクティブ・ログを使用し、ログの最後までロールフォワードを行い、停止する。 データベースはログに書き出されている最後のトランザクションがコミットされた状態にリカバリーされる。 （T5以降の時刻であれば、時刻指定でロールフォワードを停止することやEND OF BACKUP 節を使用し、最小リカバリー時間を指定してロールフォワードを停止することも可能。）	T8

4. 手順 (1/4)

- データベースに対して更新処理が実行されている状態で、データベースのオンライン・フル・バックアップを取得する。(①②)

```
$ db2 backup db SAMPLE online

バックアップは成功しました。
このバックアップ・イメージのタイム・スタンプは
20250731111027 です。
```

- このオンライン・フル・バックアップ取得中のログは、LIST HISTORY コマンドで調べることができる。以下の場合、S0000001.LOGからS0000001.LOGのログがオンライン・フル・バックアップ取得中のログである。

```
$ db2 list history backup all for SAMPLE

Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log  Backup ID
-- --- ------------------ ---- --- ------------ ------------ --------------
 B  D  20250731111027001   N    D  S0000001.LOG S0000001.LOG
--------------------------------------------------------------------------------
 5 個の表スペースを含みます:

00001 SYSCATSPACE
00002 USERSPACE1
00003 IBMDB2SAMPLEREL
00004 IBMDB2SAMPLEXML
00005 IBMDB2SAMPLEVECTOR
--------------------------------------------------------------------------------
   Comment: DB2 BACKUP SAMPLE ONLINE
(右の出力に続く)
```

```
（続き）
開始時刻: 20250731111027
   終了時刻: 20250731111028
     状況: A
 合計サイズ: 191340544 (バイト)
   シーケンス・サイズ: 191340544 (バイト)
  Compr ライブラリー:
  暗号化: いいえ
  ログの追加: はい
--------------------------------------------------------------------------------
  EID: 8 ロケーション: /home/ki1212/backup
```

4. 手順 (2/4)

- 更にデータベースに対して更新処理が実行されている状態で、データベースのオンライン増分バックアップ（デルタ)を取得する。（③④）

```
$ db2 backup db SAMPLE online incremental delta

バックアップは成功しました。
このバックアップ・イメージのタイム・スタンプは
20250731111135 です。
```

- このオンライン増分バックアップ取得中のログは、LIST HISTORY コマンドで調べることができる。以下の場合、S0000004.LOGからS0000004.LOGのログがオンライン増分バックアップ取得中のログである。

```
$ db2 list history backup all for SAMPLE

Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log  Backup ID
-- --- ------------------ ---- --- ------------ ------------ --------------
 B  D  20250731111135001   E    D  S0000004.LOG S0000004.LOG
--------------------------------------------------------------------------------
 5 個の表スペースを含みます:

00001 SYSCATSPACE
00002 USERSPACE1
00003 IBMDB2SAMPLEREL
00004 IBMDB2SAMPLEXML
00005 IBMDB2SAMPLEVECTOR
--------------------------------------------------------------------------------
   Comment: DB2 BACKUP SAMPLE ONLINE
(右の出力に続く)
```

```
（続き）
開始時刻: 20250731111135
   終了時刻: 20250731111136
     状況: A
 合計サイズ: 4775936 (バイト)
   シーケンス・サイズ: 4775936 (バイト)
  Compr ライブラリー:
  暗号化: いいえ
  ログの追加: はい
--------------------------------------------------------------------------------
  EID: 12 ロケーション: /home/ki1212/backup
```

4. 手順 (3/4)

- 更にデータベースに対して更新処理が実行されている状態で、再度データベースのオンライン増分バックアップ（デルタ）を取得する。（⑤⑥）

```
$ db2 backup db SAMPLE online incremental delta

バックアップは成功しました。
このバックアップ・イメージのタイム・スタンプは
20250731111309 です。
```

- このオンライン増分バックアップ取得中のログは、LIST HISTORY コマンドで調べることができる。以下の場合、S0000007.LOGからS0000007.LOGのログがオンライン増分バックアップ取得中のログである。

```
$ db2 list history backup all for SAMPLE

Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log  Backup ID
-- --- ------------------ ---- --- ------------ ------------ --------------
 B  D  20250731111309001   E    D  S0000007.LOG S0000007.LOG
--------------------------------------------------------------------------------
 5 個の表スペースを含みます:

00001 SYSCATSPACE
00002 USERSPACE1
00003 IBMDB2SAMPLEREL
00004 IBMDB2SAMPLEXML
00005 IBMDB2SAMPLEVECTOR
--------------------------------------------------------------------------------
   Comment: DB2 BACKUP SAMPLE ONLINE
（右の出力に続く）
```

```
（続き）
開始時刻: 20250731111309
   終了時刻: 20250731111310
     状況: A
 合計サイズ: 4775936 (バイト)
   シーケンス・サイズ: 4775936 (バイト)
  Compr ライブラリー:
  暗号化: いいえ
  ログの追加: はい
--------------------------------------------------------------------------------
  EID: 16 ロケーション: /home/ki1212/backup
```

4. 手順 (4/4)

- ここで、データベースに障害が発生し、データベースが使用不能になったことを想定する。（⑦）
- ⑤で取得したデータベースのオンライン増分バックアップ（ターゲットとなるバックアップ・イメージ）を、INCREMENTAL AUTOMATICオプションを指定してリストアする。（⑧）

```
$ db2 restore db SAMPLE incremental automatic taken at 202507311113 without prompting
DB20000I  RESTORE DATABASE コマンドが正常に完了しました。
```

- データベースに対して、ログの最後までロールフォワードを行い、停止する。 （⑨）

```
$ db2 rollforward db SAMPLE to end of logs and stop


                                ロールフォワード状況

 入力データベース別名                    = SAMPLE
 状況を返したメンバーの数                = 1

 メンバー ID                             = 0
 ロールフォワード状況                    = 非ペンディング
 次に読み込むログ・ファイル              =
 処理したログ・ファイル                  = S0000007.LOG - S0000009.LOG
 最後にコミットしたトランザクション      = 2025-07-31-02.18.51.000000 UTC

DB20000I  ROLLFORWARD コマンドが正常に完了しました。
```

- ロールフォワードの出力結果より、S0000007.LOG - S0000009.LOGが処理されたことがわかる。これは、一番最後のオンライン増分バックアップ取得中のログとそれ以降のログである。

RECOVERコマンドを使用したリカバリー

05

5. RECOVERコマンドを使用したリカバリー

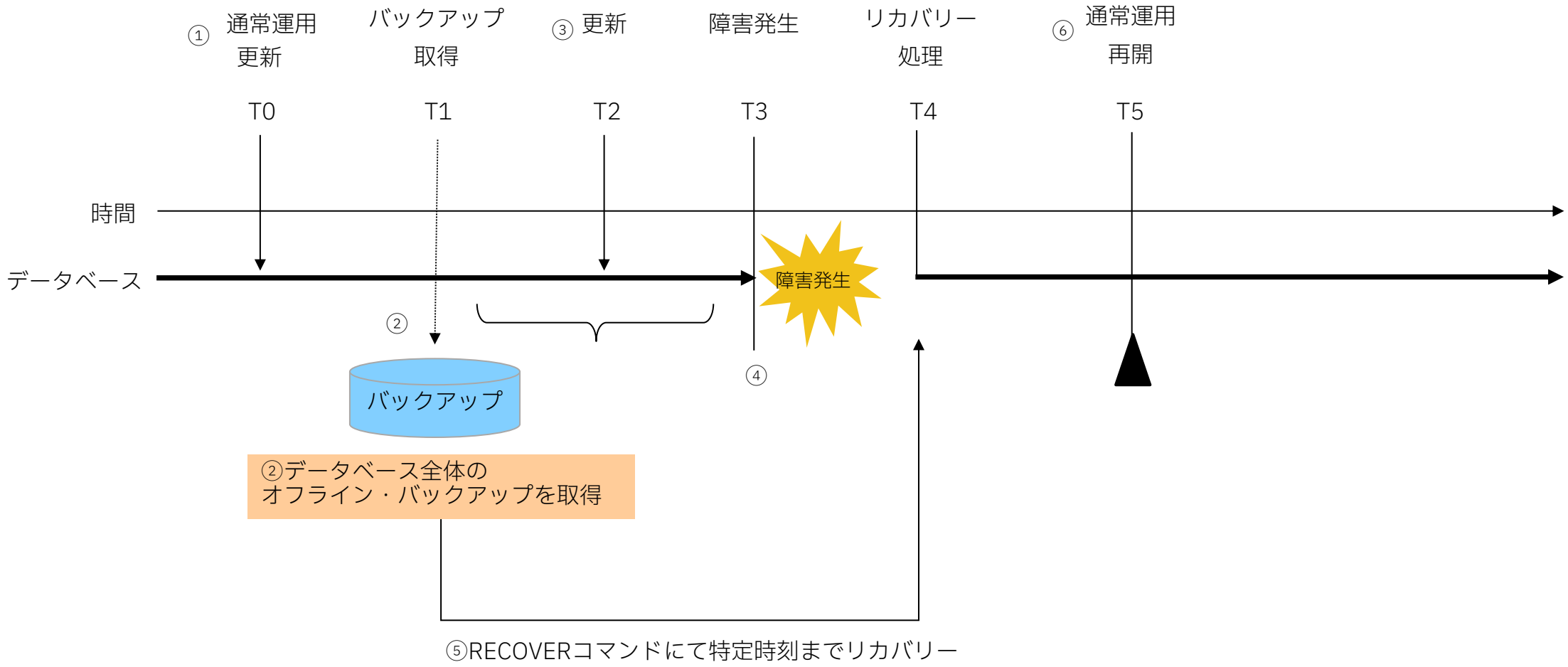
概要

– RECOVERコマンドを使用したリカバリーの手順を紹介する。
– POINT IN TIMEを使用し、特定時刻までリカバリーする。

手順

– データベースに更新を行う。
– データベースのバックアップを取得する。
– データベースの更新を行する。
– データベースを破損させる。
– RECOVERコマンドを使用して、リカバリーする。

5. リカバリーのシナリオ (1/2)



5. リカバリーのシナリオ (2/2)

	シナリオ	時刻
①	データベースに対する更新として、テーブル T1 へデータを INSERT する。 更新後に日付と時刻を確認する。	T0
②	データベースのオフライン・バックアップを取得する。	T1
③	データベースに対する更新として、テーブル T1 へデータを INSERT する。 更新後に日付と時刻を確認する。	T2
④	データベースに対する障害として、テーブル T1 のデータを削除する。 削除後に日付と時刻を確認する。	T3
⑤	RECOVERコマンドにて③の時点までリカバリーする。	T4
⑥	データベースは③の時点にリカバリーされる。	T5

5. 手順 (1/2)

- データベースに対する更新として、テーブル T1 へデータを INSERT し、日付と時刻を確認する。（①）

```
$ db2 "insert into T1 values ( 1, 'TEST' )"
DB20000I  SQL コマンドが正常に完了しました。
$ date
2025年  8月 26日 火曜日 15:54:42 JST
```

- データベース全体のオフライン・バックアップを取得する。（②）

```
$ db2 backup db SAMPLE to /work
バックアップは成功しました。このバックアップ・イメージのタイム・スタンプは 20250826155530 です。
```

- データベースに対する更新として、テーブル T1 へデータを INSERT し、日付と時刻を確認する。（③）

```
$ db2 "insert into T1 values (2, 'TEST')"
DB20000I  SQL コマンドが正常に完了しました。
$ date
2025年  8月 26日 火曜日 15:56:30 JST
```

- データベースに対する障害として、テーブル T1 のデータを削除し、日付と時刻を確認する。（④）

```
$ db2 delete from T1
DB20000I  SQL コマンドが正常に完了しました。
$ date
2025年  8月 26日 火曜日 15:57:07 JST
```

5. 手順 (2/2)

- RECOVERコマンドにて（③）の時点までリカバリーする。（⑤）

```
$ db2 recover db SAMPLE to 2025-08-26-15.56.30 using local time
                                  ロールフォワード状況
 入力データベース別名                    = SAMPLE
 状況を返したメンバーの数                  = 1
 メンバー ID                          = 0
 ロールフォワード状況                    = 非ペンディング
 次に読み込むログ・ファイル                =
 処理したログ・ファイル               = S0000008.LOG - S0000009.LOG
 最後にコミットしたトランザクション      = 2025-08-26-15.56.26.000000 Local
DB20000I  RECOVER DATABASE コマンドが正常に完了しました。
```

- （③）の時点までリカバリーされたか確認する。

```
$ db2 "select * from T1"
C1          C2
----------- ----------
          1 TEST
          2 TEST

  2 レコードが選択されました。
```

Thank you

ワークショップ、セッション、および資料は、IBMまたはセッション発表者によって準備され、それぞれ独自の見解を反映したものです。それらは情報提供の目的のみで提供されており、いかなる参加者に対しても法律的またはその他の指導や助言を意図したものではなく、またIBM製品やサービスがお客様に適用ある特定の法令に適合することを保証するものでもありません。本講演資料に含まれている情報については、完全性と正確性を期するよう努めておりますが、「現状のまま」提供され、明示または黙示にかかわらず、商業性、特定の目的への適合性、非侵害性を含め、いかなる保証も伴わないものとします。本講演資料またはその他の資料の使用によって、あるいはその他の関連によって、いかなる損害が生じた場合も、IBMは責任を負わないものとします。 本講演資料で言及されるIBM製品、プログラム、またはサービスは、IBMがビジネスを行っているすべての国・地域でご提供可能なわけではありません。

本講演資料で言及される将来の展望(製品リリース日付や製品機能を含む)は、市場機会またはその他の要因に基づいてIBM独自の決定権をもっていつでも変更できるものとし、将来の製品または機能が使用可能になること、もしくは特定の結果を確約することを意図するものではありません。本講演資料は、言及される IBM 製品またはサービスに適用ある契約条件を変更するものでも、追加の表明または保証を意図するものでもありません。

本講演資料に含まれている内容は、参加者の活動によって特定の結果が生じると述べる、または暗示することを意図したものでも、またそのような結果を生むものでもありません。 パフォーマンスは、管理された環境において標準的なIBMベンチマークを使用した測定と予測に基づいています。ユーザーが経験する実際のスループットやパフォーマンスは、ユーザーのジョブ・ストリームにおけるマルチプログラミングの量、入出力構成、ストレージ構成、および処理されるワークロードなどの考慮事項を含む、数多くの要因に応じて変化します。したがって、個々のユーザーがここで述べられているものと同様の結果を得られると確約するものではありません。記述されているすべてのお客様事例は、それらのお客様がどのようにIBM製品を使用したか、またそれらのお客様が達成した結果の実例として示されたものです。実際の環境コストおよびパフォーマンス特性は、お客様ごとに異なる場合があります。