

# Oracle Direct Seminar



**ORACLE®**

**実践!! バックアップ・リカバリ**

**ー論理障害から復旧するためのフラッシュバック大全ー**

日本オラクル株式会社

**Oracle Direct**



# Agenda

- フラッシュバック機能とは
- 参照系機能
  - フラッシュバック問合せ
  - フラッシュバック・バージョン問合せ
  - フラッシュバック・トランザクション問合せ
  - フラッシュバック・データ・アーカイブ
- リカバリ系機能
  - フラッシュバック・トランザクション
  - フラッシュバック・テーブル
  - フラッシュバック・ドロップ
  - フラッシュバック・データベース

## Oracle Directの無償技術サービス

- SQL Serverからの移行アセスメント
- MySQLからの移行相談
- PostgreSQLからの移行相談
- Accessからの移行アセスメント
- Oracle Application Serverからの移行相談
- Oracle Database バージョンアップ支援
- WebLogic Serverバージョンアップ支援
- Oracle Application Serverからの移行相談
- Oracle Developer/2000 Webアップグレード相談
- パフォーマンス・クリニック
- Webシステム・ボトルネック診断サービス
- Oracle Database 構成相談
- システムセキュリティ診断

<http://www.oracle.com/lang/jp/direct/services.html>

# あらゆる障害ケースへの対応策

- 事業継続計画ではあらゆる障害ケースへの対応策が不可欠
- High Availability: システムを24時間365日とまらずに運用する
  - サービスを継続させるために
    - ハードウェア、ソフトウェアの障害発生時にもサービス提供を継続
      - Real Application Clusters
      - Automatic Storage Management
      - Data Guard
  - 障害時にもダウンタイムを最小限に抑えるために
    - 様々な障害パターンからの迅速なリカバリ
      - 物理障害
      - 論理障害

# 物理障害と論理障害

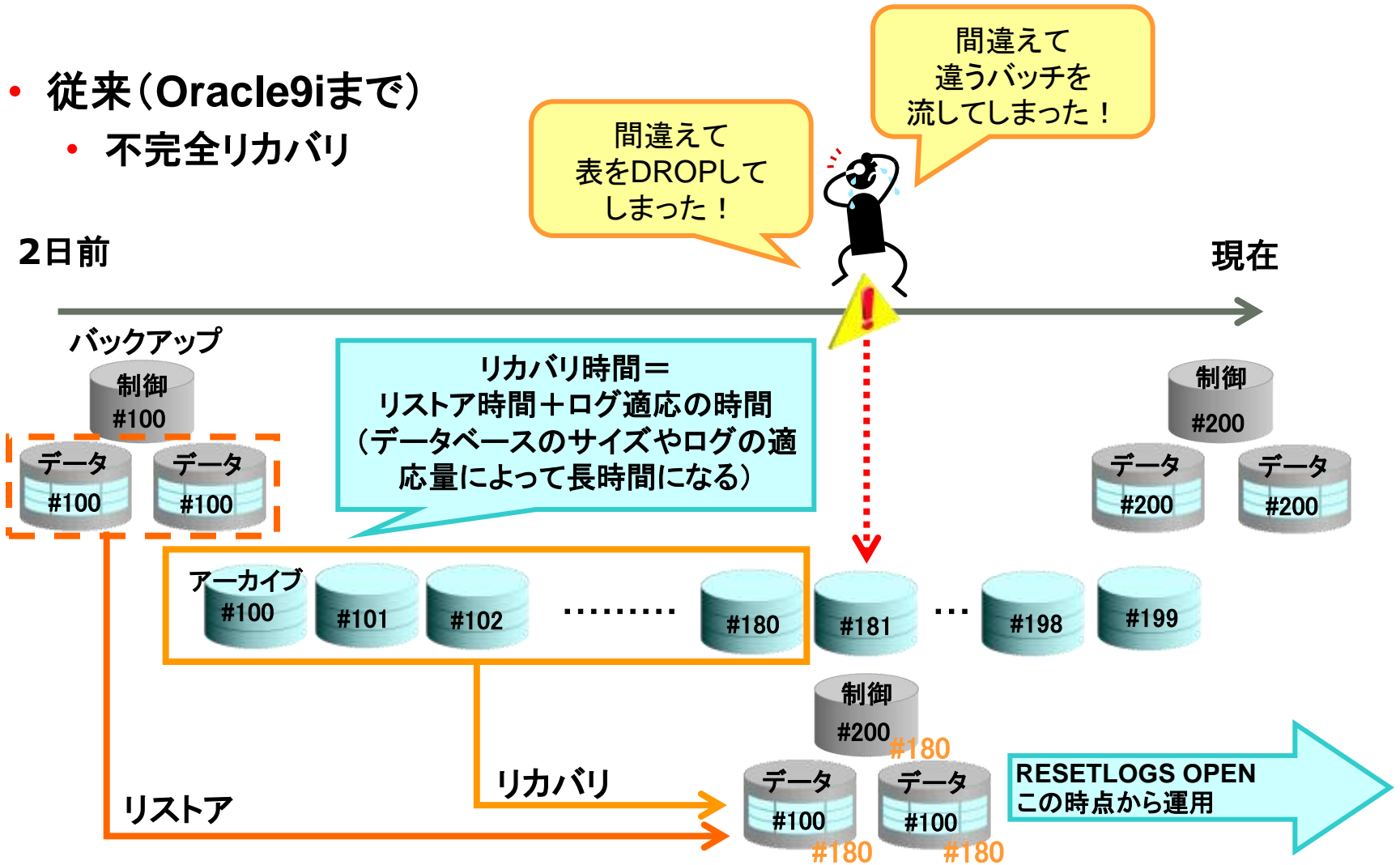
- **物理障害: データベース構成ファイルの破損**
  - 対策
    - ハードウェアのミラーリング
    - 定期バックアップ
- **論理障害: ユーザーやアプリからの誤った変更**
  - 人が関わる部分のため、発生を押さえることは困難
    - 「違うバッチ流してしまった！！」
    - 「違うテーブルを DROP してしまった！」
    - 「3時間前の データベースのデータを知りたい！」



アプリケーション障害の40%は  
オペレータやユーザーのエラー

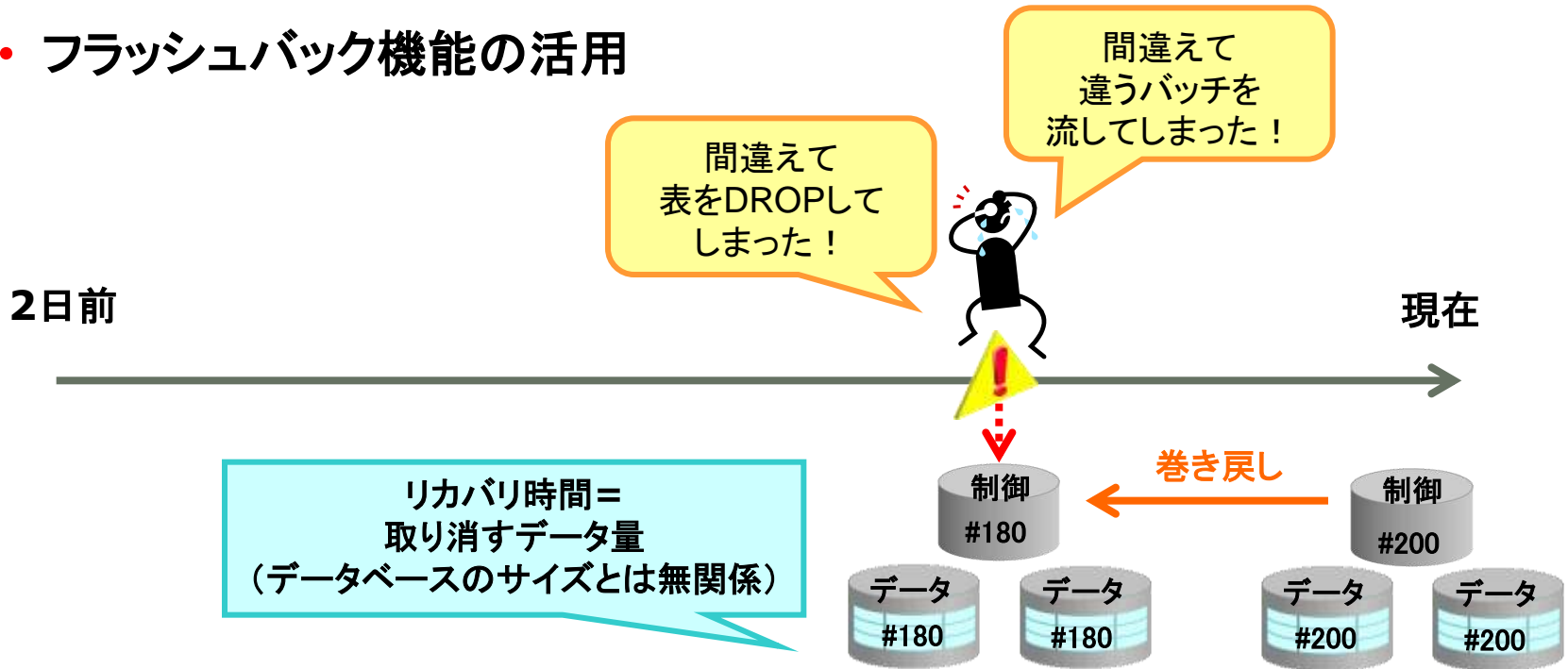
# 論理障害からの復旧<従来>

- 従来 (Oracle9iまで)
  - 不完全リカバリ



# 論理障害からの復旧<フラッシュバック>

- フラッシュバック機能の活用



バックアップからリストアすることなく、復旧可能  
リストア時間が不要なので、従来の不完全リカバリに比べて簡単+高速

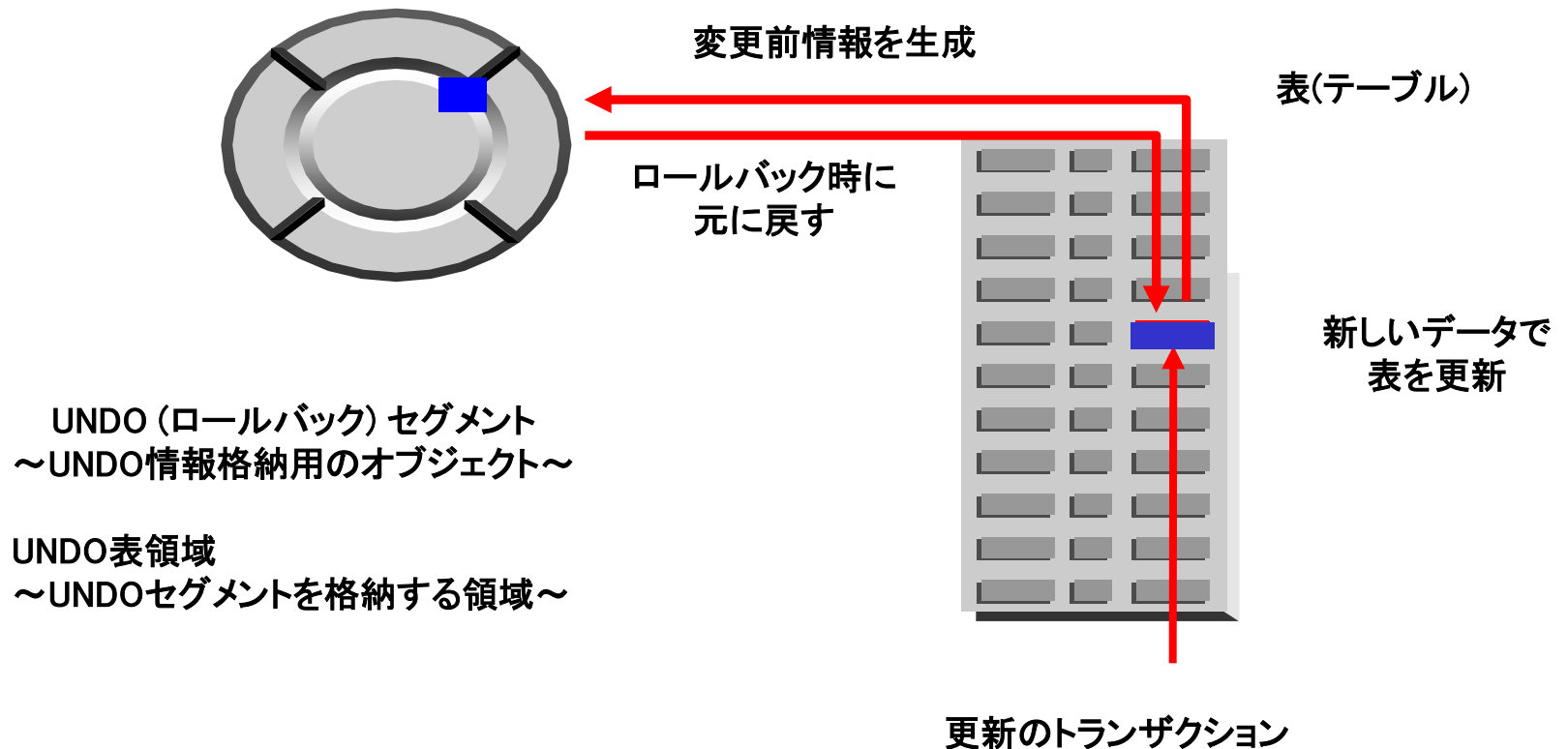
# フラッシュバック機能とは

- **論理障害に特化したリカバリ機能**
  - 人為的(ユーザー)エラーからの早急な復旧
    - 重要なデータの削除、間違ったデータの更新、間違った表の削除など、メンテナンス・ミスによる問題
    - アプリケーションミス、誤バッチの実行による問題
- **データベース全体、または一部を、過去の状態へと「巻き戻す」機能**
  - バックアップからのリカバリが不要
    - 従来の不完全リカバリに比べ非常に簡単
    - 短時間で復旧
  - 過去データの参照が可能
    - 不正なデータ改竄に効果

**論理障害による業務(システム)停止を最低限に抑えることが可能**

# フラッシュバック機能のための設定(UNDOデータ)

- データを元に戻すための「ログ」が必要
  - 多くのフラッシュバック機能ではUNDOデータを使用
  - UNDO以外の設定が必要なフラッシュバック機能もある





# フラッシュバック機能のための設定(UNDOデータ)

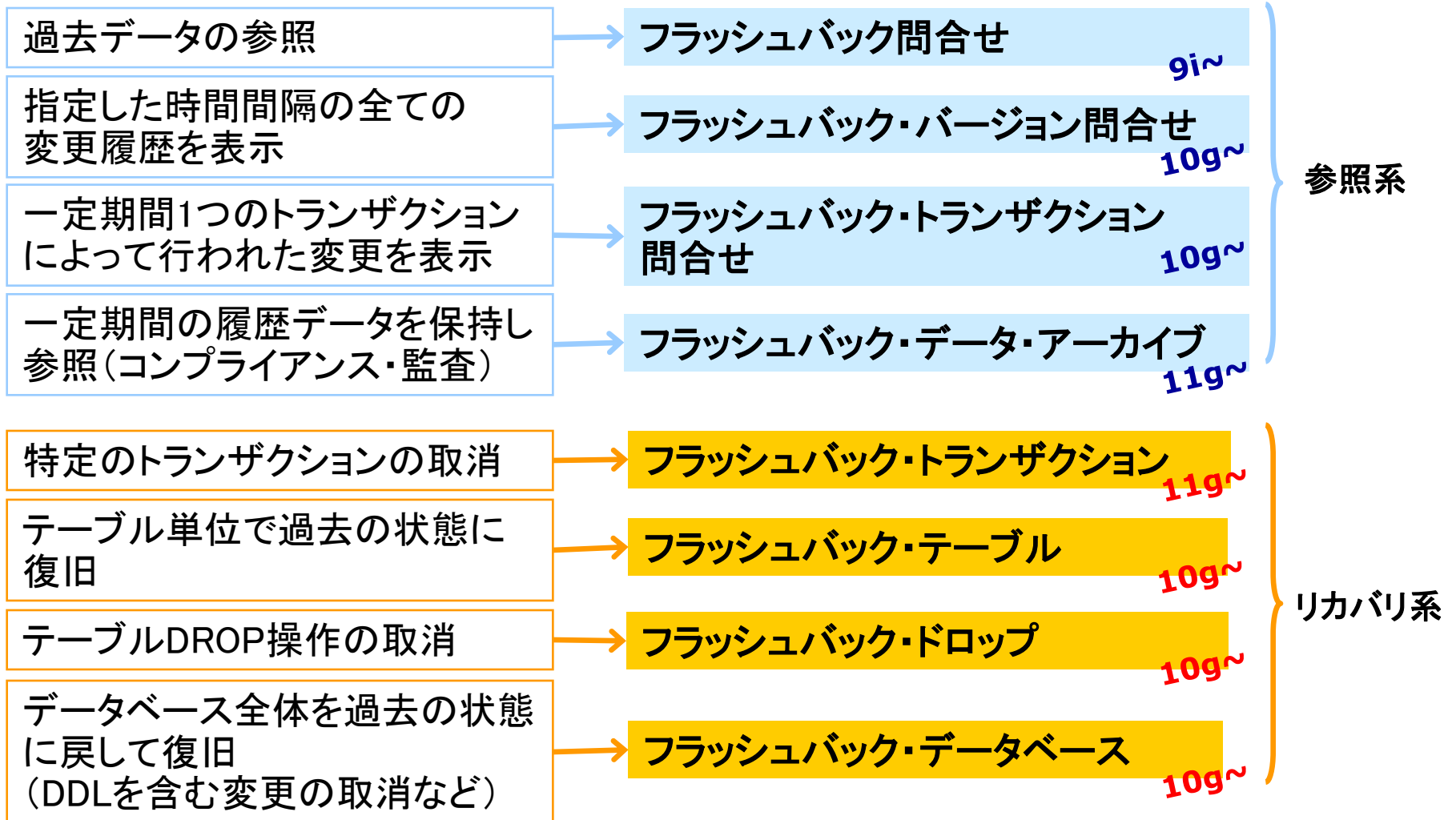
## UNDO(変更前データ)の管理

- UNDO表領域は循環運用(古いものは上書きされる)
  - UNDOデータが上書きされると、フラッシュバックは不可
- 要件にあったUNDO表領域の設定が重要



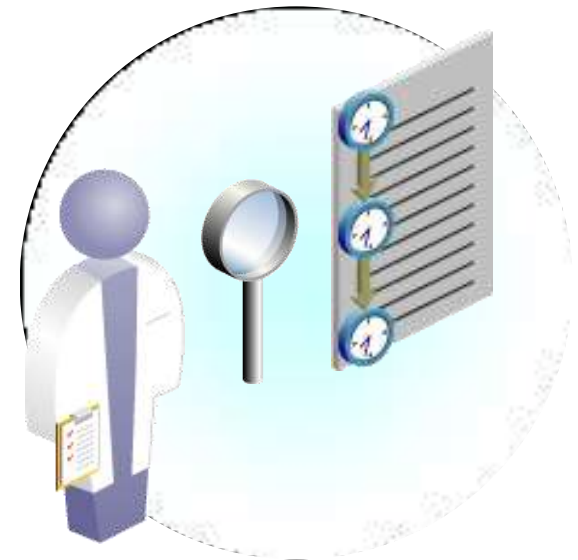
- 8i以前: 手動管理
- 9i以降: 自動UNDO管理
  - UNDO\_MANAGEMENT: 自動UNDO領域管理モード
  - UNDO\_TABLESPACE: 使用するUNDO表領域
  - UNDO\_RETENTION: 保持するUNDO情報量(秒単位)を指定
- 10g以降: 自動UNDOチューニング機能
  - 表領域のサイズやランザクションの長さによって、動的にUNDO保持期間が変更される

# 様々なフラッシュバック機能



# Agenda

- フラッシュバック機能とは
- 参照系機能
  - フラッシュバック問合せ
  - フラッシュバック・バージョン問合せ
  - フラッシュバック・トランザクション問合せ
  - フラッシュバック・データ・アーカイブ
- リカバリ系機能
  - フラッシュバック・トランザクション
  - フラッシュバック・テーブル
  - フラッシュバック・ドロップ
  - フラッシュバック・データベース

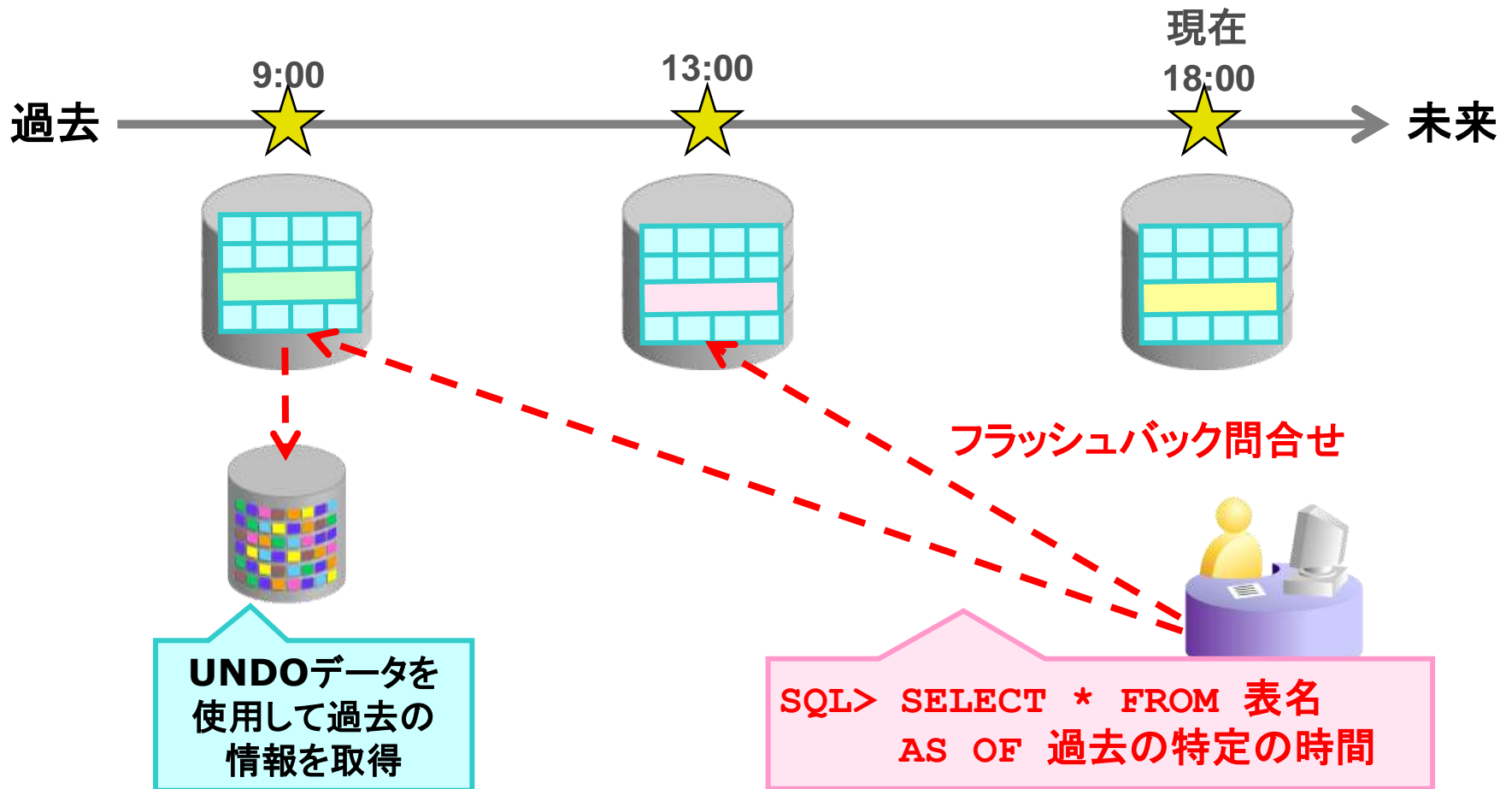


# 参照系のフラッシュバック機能

- 過去のデータを参照するためのフラッシュバック機能
    - フラッシュバック問合せ (9i~)
      - 過去の一時店を指定し、その時点のデータを参照
    - フラッシュバック・バージョン問合せ (10g~)
      - 特定の時間間隔を指定し、変更履歴を追跡
    - フラッシュバック・トランザクション問合せ (10g~)
      - トランザクション単位での変更履歴を表示
    - フラッシュバック・データ・アーカイブ (11g~)
      - 特定の表に対して、より長期間の履歴保存を可能にする設定
- UNDOデータを利用

# フラッシュバック問合せ

- SELECT文のAS OF句で指定した過去のある時点のデータを取得



# フラッシュバック問合せ 実行例

- フラッシュバック問合せ実行例
  - 特定の時間を指定して過去のデータを参照

```
SQL> SELECT * FROM employees
      AS OF TIMESTAMP TO_TIMESTAMP
      ('2009-02-14 09:00:00', 'YYYY-MM-DD HH:MI:SS');
```

- 「何分前」という指定で過去のデータを参照

```
SQL> SELECT * FROM employees
      AS OF TIMESTAMP
      (SYSTIMESTAMP - INTERVAL '60' MINUTE);
```

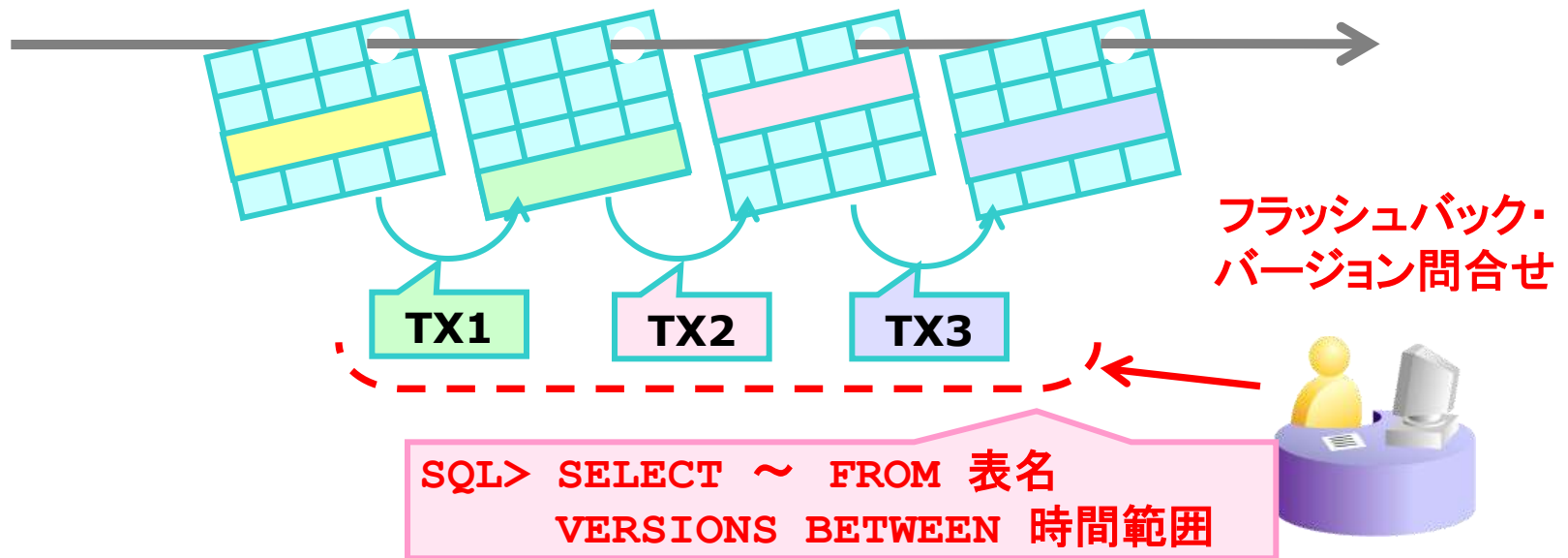
- SCN(データベースの時間を管理するための内部的な一意の順序番号)を指定して過去のデータを参照

```
SQL> SELECT * FROM employees
      AS OF SCN(496797);
```

# フラッシュバック・バージョン問合せ

- SELECT文でVERSIONS BETWEEN句を使用して、フラッシュバック・バージョン問合せを指定
- 特定の時間間隔内に存在していた、特定の行の様々なバージョンを取得  
→ **どういう経緯で現在の値に至ったのか確認可能**

トランザクションが実行され、**COMMIT**文が実行されると、  
そのたびに新しい行バージョンが作成



# フラッシュバック・バージョン問合せ 実行例

- DEPT表に対して3つのトランザクションを実行

```
SQL> SELECT * FROM dept;
DEPTNO  DNAME          LOC
-----  -
10      ACCOUNTING     NEW YORK
20      RESEARCH       DALLAS
30      SALES          CHICAGO
40      OPERATIONS     BOSTON

SQL> INSERT INTO dept VALUES (50, 'DIRECT', 'TOKYO');
SQL> commit;

SQL> UPDATE dept SET loc='OSAKA' WHERE deptno=50;
SQL> commit;

SQL> UPDATE dept SET loc = 'FUKUOKA' WHERE deptno=50;
SQL> commit;
```



# フラッシュバック・バージョン問合せ 実行例(続き) <sup>10g~</sup>

- フラッシュバック・バージョン問合せでトランザクションを確認

```
SQL> SELECT  versions_xid, versions_starttime,
             versions_operation, deptno, loc
           FROM    dept
             VERSIONS BETWEEN scn minvalue and maxvalue;
```

「UNDOに残っている  
全て」という指定  
明示的に時間を指定  
することも可能

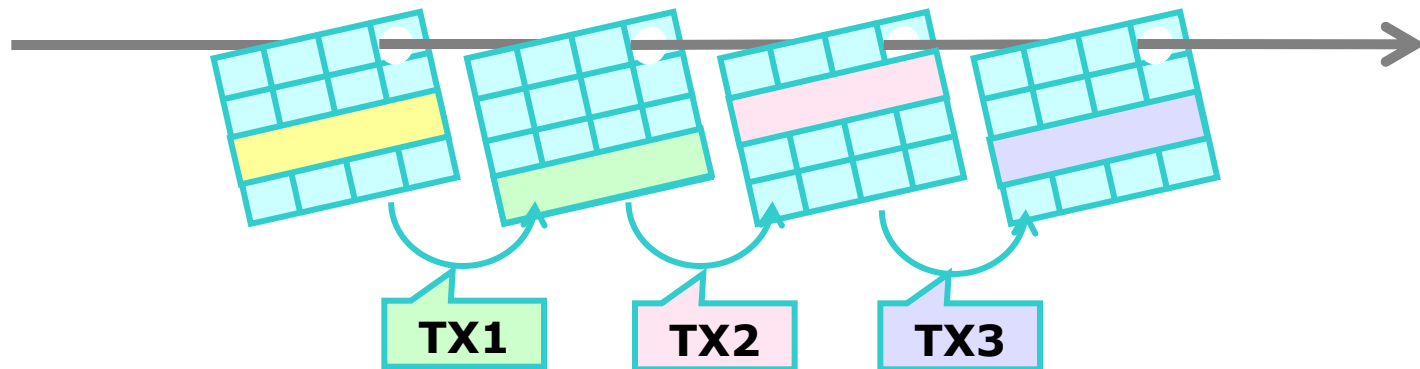
VERSIONS_XID	VERSIONS_STARTTIME	VER	LOC
-----	-----	-----	-----
		10	NEW YORK
		20	DALLAS
		30	CHICAGO
		40	BOSTON
050005009A030000	09-02-13 15:42:10	U	50 FUKUOKA
04001D0093030000	09-02-13 15:40:50	U	50 OSAKA
03001600B3030000	09-02-13 15:36:27	I	50 TOKYO

この変更を行った  
トランザクションID

UPDATE=U  
DELETE=D  
INSERT =I

# フラッシュバック・トランザクション問合せ

- トランザクション単位での変更履歴を表示
  - 1トランザクションで複数表や行を変更した場合の追跡に有効
  - 変更を取り消すSQL文 (UNDO SQL文) を取得可能



```
SQL>SELECT * FROM FLASHBACK_TRANSACTION_QUERY  
WHERE XID='XXXXXXXXXXXXX';
```

トランザクションIDを指定して、このトランザクション  
によって行われた変更を追跡



# フラッシュバック・トランザクション問合せ 実行例

- フラッシュバック・トランザクションでUNDO SQLを取得

## フラッシュバック・バージョン問合せの結果

VERSIONS_XID	VERSIONS_STARTTIME	VER	LOC
		10	NEW YORK
		20	DALLAS
		30	CHICAGO
		40	BOSTON
050005009A030000	09-02-13 15:42:10	U 50	FUKUOKA
04001D0093030000	09-02-13 15:40:50	U 50	OSAKA
03001600B3030000	09-02-13 15:36:27	I 50	TOKYO

フラッシュバック・バージョン問合せで調べたトランザクションIDを指定して、特定のトランザクションでの処理を追跡+UNDO SQLを取得

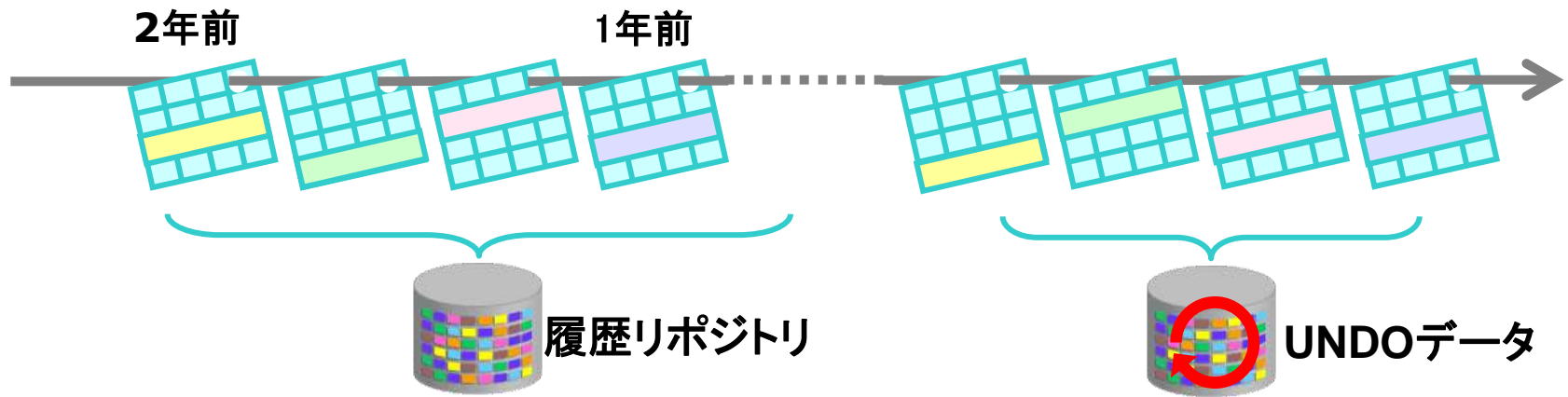
```
SQL> SELECT operation,undo_sql
FROM flashback_transaction_query
WHERE xid='04001D0093030000';
```

```
OPERATION
-----
UNDO_SQL
-----
UPDATE
UPDATE "DEPT" SET "LOC" = 'TOKYO' WHERE ROWID = 'AAAREkAAEAAAActAAA';
```

この処理を取り消すためのSQL文  
 INSERT→DELETE  
 UPDATE→UPDATE  
 DELETE→INSERT

# フラッシュバック・データ・アーカイブ ( Total Recall ) <sup>11g~</sup>

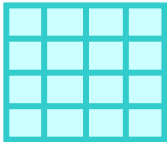
- より長期間保存可能な更新データの履歴リポジトリ
  - UNDOデータを専用の領域に保存することにより、確実に保存
  - 特定の表に対して、履歴保存の設定が可能
- 使用用途
  - コンプライアンス
  - 監査レポート
  - Information Lifecycle Management (ILM)
  - データ解析



# フラッシュバック・データ・アーカイブの設定

- 表領域にフラッシュバック・データ・アーカイブ用の領域を用意
  - 適切な保存期間を設定
  - 保存期間ごとに複数の領域を作成可能
- 履歴情報を残したい表に対してフラッシュバック・データ・アーカイブを有効化

特定の表



```
SQL>CREATE FLASHBACK ARCHIVE fda_test  
TABLESPACE 表領域  
QUOTA 50M RETENTION 1 YEAR;
```

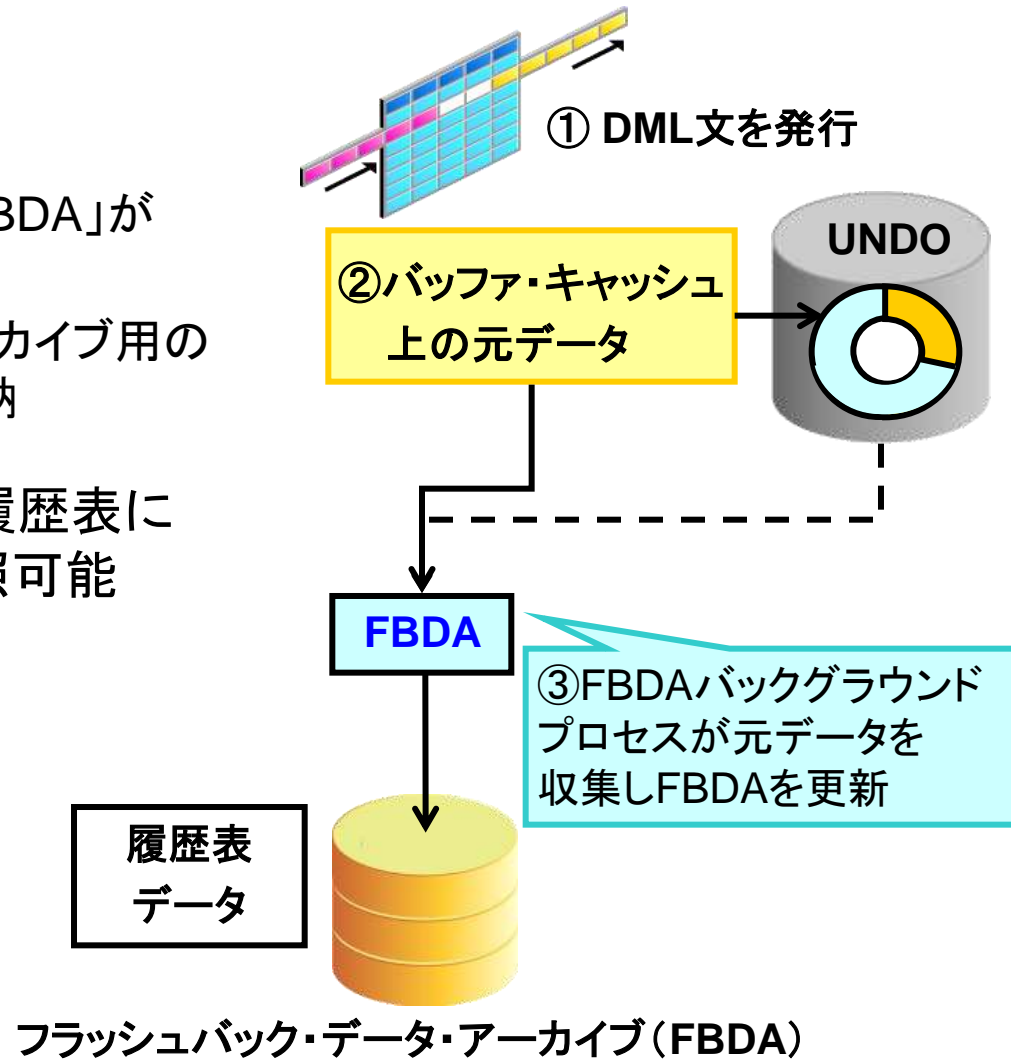
```
SQL>ALTER TABLE 表名  
FLASHBACK ARCHIVE fda_test;
```



履歴表  
データ

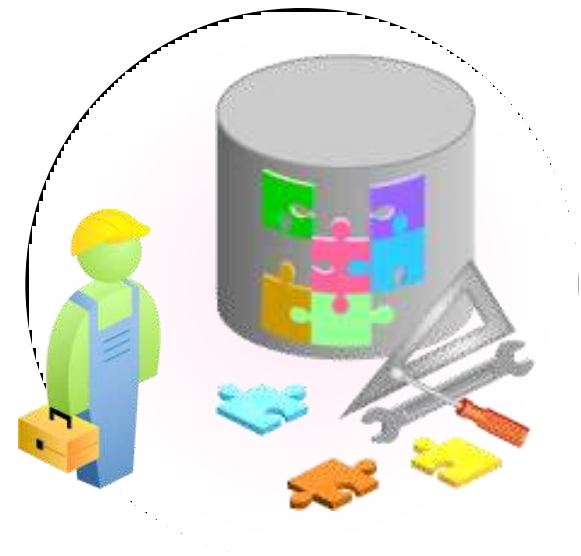
# フラッシュバック・データ・アーカイブのしくみ

- ログ保存のしくみ
  - ①表のデータ更新
  - ②バックグラウンド・プロセス「FBDA」が UNDOデータを集集
  - ③フラッシュバック・データ・アーカイブ用の履歴表にUNDOデータを格納
- 通常のフラッシュバック文で、履歴表に残っている期間のデータを参照可能



# Agenda

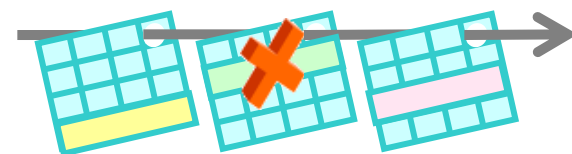
- フラッシュバック機能とは
- 参照系機能
  - フラッシュバック問合せ
  - フラッシュバック・バージョン問合せ
  - フラッシュバック・トランザクション問合せ
  - フラッシュバック・データ・アーカイブ
- リカバリ系機能
  - フラッシュバック・トランザクション
  - フラッシュバック・テーブル
  - フラッシュバック・ドロップ
  - フラッシュバック・データベース



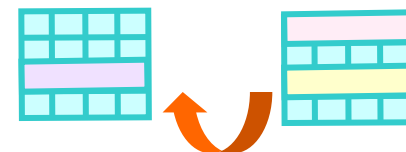
# リカバリ系のフラッシュバック機能

- データを過去の状態に戻すためのフラッシュバック機能

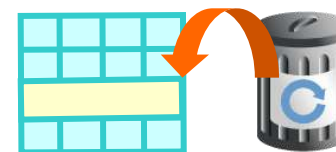
- フラッシュバック・トランザクション(11g~)
  - 誤って実行したトランザクションの取り消し



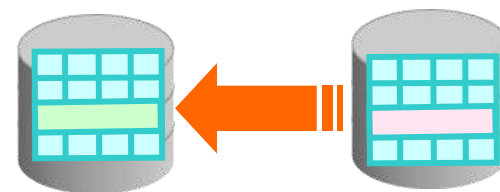
- フラッシュバック・テーブル(10g~)
  - 表に対して誤って実行したDML処理の取り消し
  - 表単位で過去の状態に戻す



- フラッシュバック・ドロップ(10g~)
  - 誤って実行したDROPの取り消し



- フラッシュバック・データベース(10g~)
  - データベース全体を過去の状態に戻す

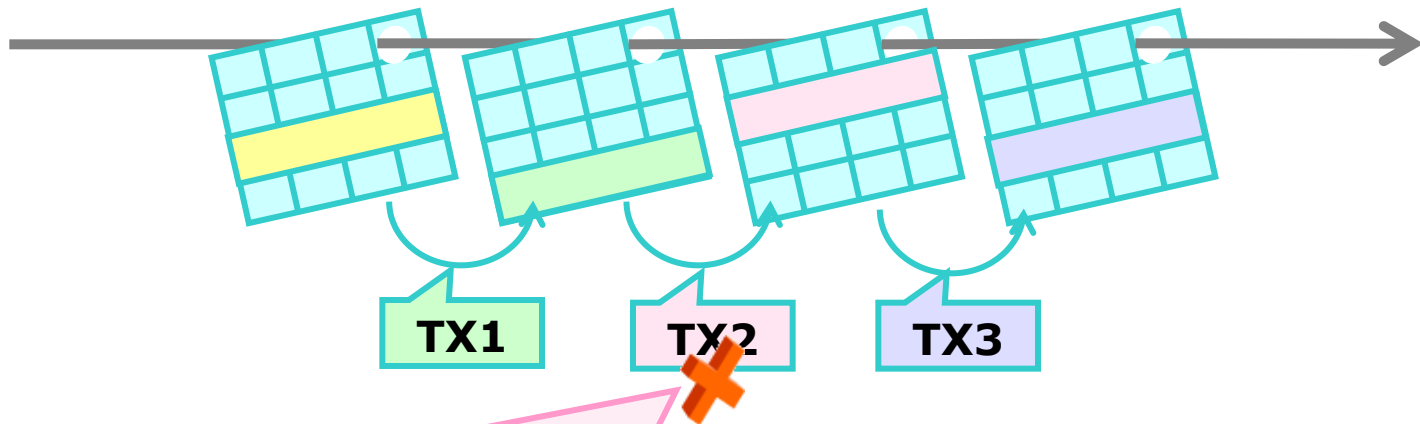




# フラッシュバック・トランザクション

11g~

- 誤って実行したトランザクションの取り消し
  - トランザクション識別子 (XID) を指定して、特定のトランザクションを取り消し



DBMS\_FLASHBACK.TRANSACTION\_BACKOUT プロシージャで  
トランザクションIDを指定して、特定のトランザクションを取り消し

# フラッシュバック・トランザクションのための設定

11g~

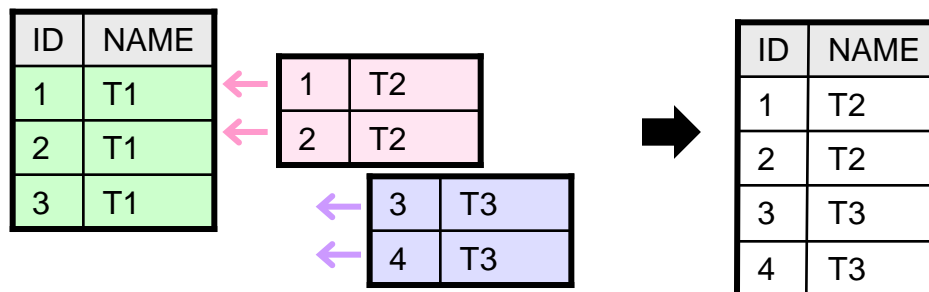
- フラッシュバック・トランザクションの仕組み
  - REDOログの情報を使用して、特定のトランザクションによる変更を取り消し
    - オンラインREDOログ、アーカイブREDOログを使用可能
    - サプリメンタル・ロギングを有効化する必要

```
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;  
  
SQL> SELECT supplemental_log_data_min FROM v$database;  
SUPPLEME  
-----  
YES
```

サプリメンタル・ロギング:  
本来REDOログ情報(更新前と更新後の値)  
に加えて補助的な情報(更新列以外の列値)を  
追加で残す設定

# フラッシュバック・トランザクション 実行例

- 3つのトランザクションを実行



```
SQL> CREATE table test( id number, name varchar2(5));
```

```
SQL> INSERT INTO test VALUES (1, 'T1');
```

```
SQL> INSERT INTO test VALUES (2, 'T1');
```

```
SQL> INSERT INTO test VALUES (3, 'T1');
```

```
SQL> commit;
```

トランザクション1

```
SQL> UPDATE test SET name='T2' WHERE id=1;
```

```
SQL> UPDATE test SET name='T2' WHERE id=2;
```

```
SQL> commit;
```

トランザクション2

```
SQL> UPDATE test SET name='T3' WHERE id=3;
```

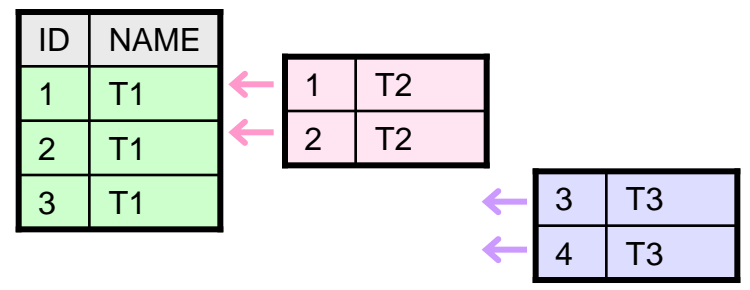
```
SQL> INSERT INTO test VALUES (4, 'T3');
```

```
SQL> commit;
```

トランザクション3

# フラッシュバック・トランザクション 実行例(続き)

- トランザクションIDを確認  
(フラッシュバック・バージョン問合せを使用)



```
SQL> SELECT versions_xid,versions_operation, id,name
2 FROM test
3 VERSIONS BETWEEN scn minvalue and maxvalue;
```

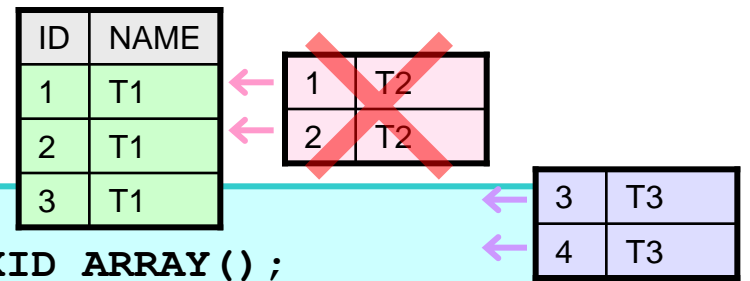
VERSIONS_XID	V	ID	NAME
04001A0043030000	I	4	T3
04001A0043030000	U	3	T3
08000D00A7030000	U	2	T2
08000D00A7030000	U	1	T2
02000B00A0030000	I	3	T1
02000B00A0030000	I	2	T1
02000B00A0030000	I	1	T1

フラッシュバック・バージョン問合せでトランザクションIDを特定

トランザクション3  
トランザクション2  
トランザクション1

# フラッシュバック・トランザクション 実行例(続き)

- フラッシュバック・トランザクション実行



```
SQL> DECLARE
  2   f_xid sys.XID_ARRAY := sys.XID_ARRAY();
  3   BEGIN
  4   f_xid.extend;
  5   f_xid(1) := HEXTORAW('08000D00A7030000');
  6   DBMS_FLASHBACK.TRANSACTION_BACKOUT(1, f_xid);
  7   END;
  8   /
```

PL/SQLプロシージャが正常に完了しました。

```
SQL> SELECT * FROM test;
      ID NAME
-----
      1 T1
      2 T1
      3 T3
      4 T3

SQL> commit;
```

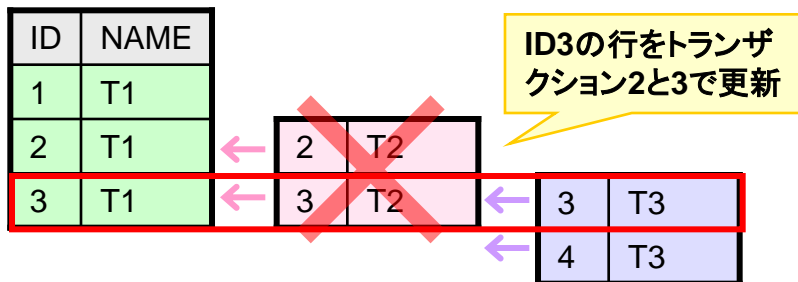
フラッシュバック・バージョン問合せで  
特定したトランザクションIDを指定して  
このトランザクションを取り消し

フラッシュバックは  
COMMITによって  
確定(ROLLBACK  
で取り消し可能)

# フラッシュバック・トランザクションのオプション指定

11g~

- トランザクションに依存関係があった場合



```
SQL> DECLARE
  2   f_xid sys.XID_ARRAY := sys.XID_ARRAY();
  3   BEGIN
  4   f_xid.extend;
  5   f_xid(1) := HEXTORAW('03002000B9030000');
  6   DBMS_FLASHBACK.TRANSACTION_BACKOUT(1,f_xid);
  7   END;
  8   /
```

DECLARE

\*

行1でエラーが発生しました。:

ORA-55504: トランザクションがNOCASCADEモードで競合しています

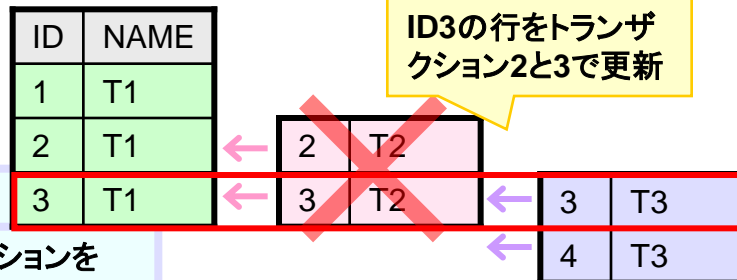
ORA-06512: "SYS.DBMS\_FLASHBACK", 行37

ORA-06512: "SYS.DBMS\_FLASHBACK", 行70

ORA-06512: 行6

# フラッシュバック・トランザクションのオプション指定 11g~

- トランザクションに依存関係があった場合のフラッシュバック動作設定



<b>NOCASCADE</b>	依存性が検出された場合は、エラー(デフォルト)
<b>NOCASCADE_FORCE</b>	依存トランザクションを考慮せずに、指定したトランザクションを強制的に取り消す
<b>NONCONFLICT_ONLY</b>	競合が発生していない行に対する変更のみを取り消す
<b>CASCADE</b>	指定したトランザクションおよび依存するトランザクションを後順(コミット時の逆の順序)で完全に取り消す

```
SQL> DECLARE
  2   f_xid sys.XID_ARRAY := sys.XID_ARRAY();
  3 BEGIN
  4   f_xid.extend;
  5   f_xid(1) := HEXTORAW('03002000B9030000');
  6   DBMS_FLASHBACK.TRANSACTION BACKOUT
  7   (1, f_xid, DBMS_FLASHBACK.NONCONFLICT_ONLY);
  8 END;
  9 /

SQL> SELECT * FROM test;

  ID NAME
-----
  1 T1
  2 T1
  3 T3
  4 T3
```

トランザクション3と競合したID3は取り消されず、ID2だけ取り消された

```
SQL> DECLARE
  2   f_xid sys.XID_ARRAY := sys.XID_ARRAY();
  3 BEGIN
  4   f_xid.extend;
  5   f_xid(1) := HEXTORAW('03002000B9030000');
  6   DBMS_FLASHBACK.TRANSACTION BACKOUT
  7   (1, f_xid, DBMS_FLASHBACK.NOCASCADE_FORCE);
  8 END;
  9 /

SQL> SELECT * FROM test;

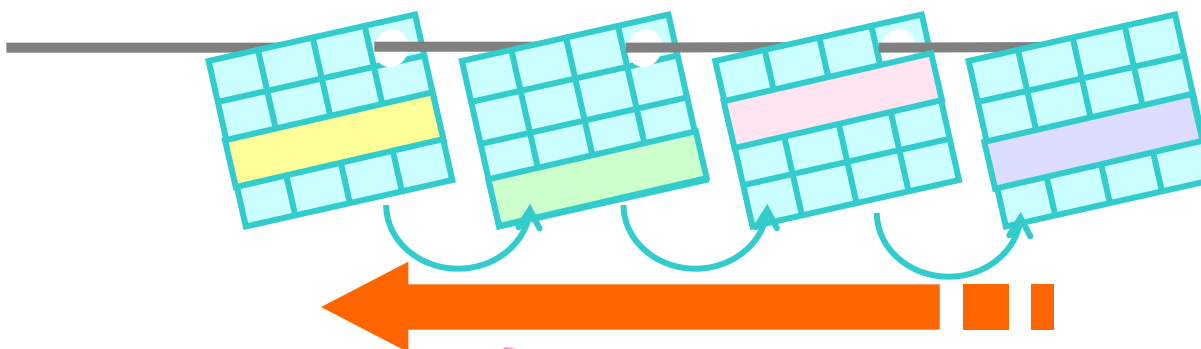
  ID NAME
-----
  1 T1
  2 T1
  3 T1
  4 T3
```

トランザクション3を無視して、トランザクション2が取り消された

# フラッシュバック・テーブル

10g~

- 特定のテーブルデータだけを指定時点の状態に戻す機能
  - 誤って実行したDML文の取り消しが可能
- 制限事項
  - 表構造の変更はフラッシュバック不可
  - TRUNCATEを含む全DDL文は取り消し不可



```
SQL>FLASHBACK TABLE 表名 TO 時間;
```



# フラッシュバック・テーブルのための設定

- フラッシュバック・テーブルの仕組み
  - UNDOの情報を使用して、表を特定の時間に戻す
  - 内部的なDML文を発行してデータに戻す
    - フラッシュバック操作前に「行移動の有効化」の設定をする必要
    - ROWIDの値が保証されなくなるので、ROWIDを使用したSQLを使用している場合は考慮が必要

```
SQL> FLASHBACK TABLE dept to timestamp  
      (systimestamp-interval '2' minute);  
FLASHBACK TABLE dept to timestamp (systimestamp-interval '2' minute)
```

\*

行1でエラーが発生しました。:

ORA-08189: 行移動が使用できないため、表をフラッシュバックできません

「行移動の有効化」を  
しておかないと、  
フラッシュバック操作  
時にエラーが発生

```
SQL> ALTER TABLE dept ENABLE ROW MOVEMENT;
```

# フラッシュバック・テーブル 実行例

- フラッシュバック・テーブル実行例

```
SQL> SELECT * FROM dept;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

```
SQL> DELETE FROM dept;
```

```
SQL> commit;
```

```
SQL> select * from dept;
```

レコードが選択されませんでした。

```
SQL> ALTER TABLE dept ENABLE ROW MOVEMENT;
```

```
SQL> FLASHBACK TABLE dept to timestamp  
(systimestamp-interval '2' minute);
```

```
SQL> SELECT * FROM dept;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

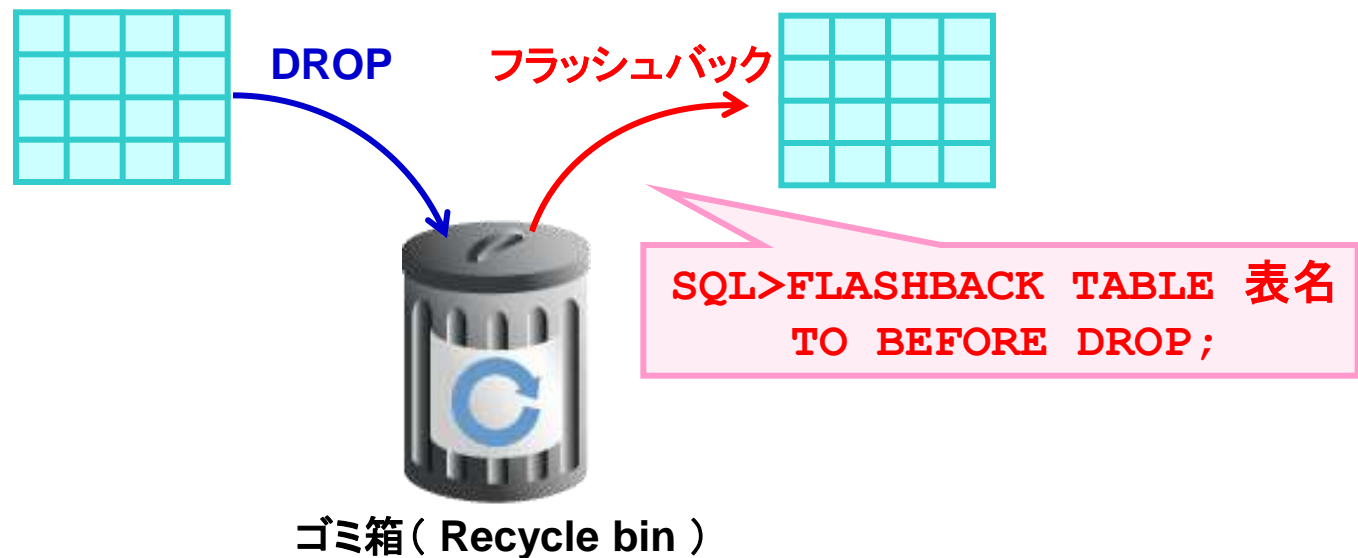
「2分前」を指定して  
フラッシュバック

「〇時〇分」やSCN  
指定でも可能

# フラッシュバック・ドロップ

10g~

- 削除した表を元に戻す機能
  - 誤って実行したDROPの取り消しが可能
- 制限事項
  - 表領域ごと削除した場合は、フラッシュバック不可
  - システム表領域を除く、ローカル管理表領域に対してのみ使用可能



# フラッシュバック・ドロップ 実行例

- 10g以降、オブジェクトをDROPで削除すると、「ゴミ箱」に入る
  - ゴミ箱に入ったオブジェクトは「BIN\$~」という名前に改名
  - 表とすべての依存オブジェクト(索引、トリガーなど)もゴミ箱に移動し、フラッシュバックを実行すると、すべてのオブジェクトが同時に復旧

```
SQL> DROP TABLE dept;
```

```
SQL> SELECT object_name, original_name, operation, createtime
       FROM recyclebin;
```

OBJECT_NAME	ORIGINAL_N	OPERATION	CREATETIME
BIN\$99KyJrI2SVCWdILDFh670A==\$0 DEPT PK		DROP	2009-02-12:17:43:26
BIN\$SjP67teUSb+VKCBc3uLzCg==\$0 DEPT		DROP	2009-02-12:17:43:16

DEPT表と索引が  
ゴミ箱に入っている

```
SQL> FLASHBACK TABLE dept TO BEFORE DROP;
```

# フラッシュバック・ドロップの注意点

- フラッシュバック時のオブジェクト名
  - 表をフラッシュバックする際、表名を変えてフラッシュバックすることが可能

すでに別の「DEPT表」がある場合などに有効

```
SQL> FLASHBACK TABLE dept TO BEFORE DROP RENAME TO old_dept;
```

- フラッシュバック実行時、表の名前は元の名前に戻るが、依存オブジェクトは元の名前に戻らないため、改名したほうがよい

```
SQL> FLASHBACK TABLE dept TO BEFORE DROP;
```

```
SQL> SELECT object_name, object_type FROM user_objects;
```

OBJECT_NAME	OBJECT_TYPE
DEPT	TABLE
BIN\$SjP67teUSb+VKCBc3uLzCg==\$0	INDEX

```
SQL> ALTER INDEX "BIN$SjP67teUSb+VKCBc3uLzCg==$0" TO dept_idx;
```

=などは通常のオブジェクト名にはできないので、“ ”が必要

# フラッシュバック・データベース

10g~

- データベース全体を指定時点の状態に戻す機能
  - 従来の不完全リカバリに代わる機能  
ただしデータベース構成ファイルの物理的破損には対応しない
- 制限事項
  - データベースを停止する必要
  - RESETLOGSが必要(不完全リカバリと同様)
  - フラッシュバック・ログをとっておく必要

間違えたバッチを流し、データがぐちゃぐちゃになった！  
表をTRUNCATEした！  
間違えてユーザーを削除した！



ORACLE

# フラッシュバック・データベースのための設定

- ARCHIVE LOGモードの設定
- フラッシュバック・データベース・ログの格納先(フラッシュリカバリ領域)の設定
  - フラッシュリカバリ領域の場所とサイズを指定
  - フラッシュバック・データベース・ログの保持期間

## 初期化パラメータ

```
DB_RECOVERY_FILE_DEST='/u01/app/oracle/flash_recovery_area'  
DB_RECOVERY_FILE_DEST_SIZE=10G  
DB_FLASHBACK_RETENTION_TARGET=1440
```

フラッシュバック・データベース・  
ログの保持期間(分)

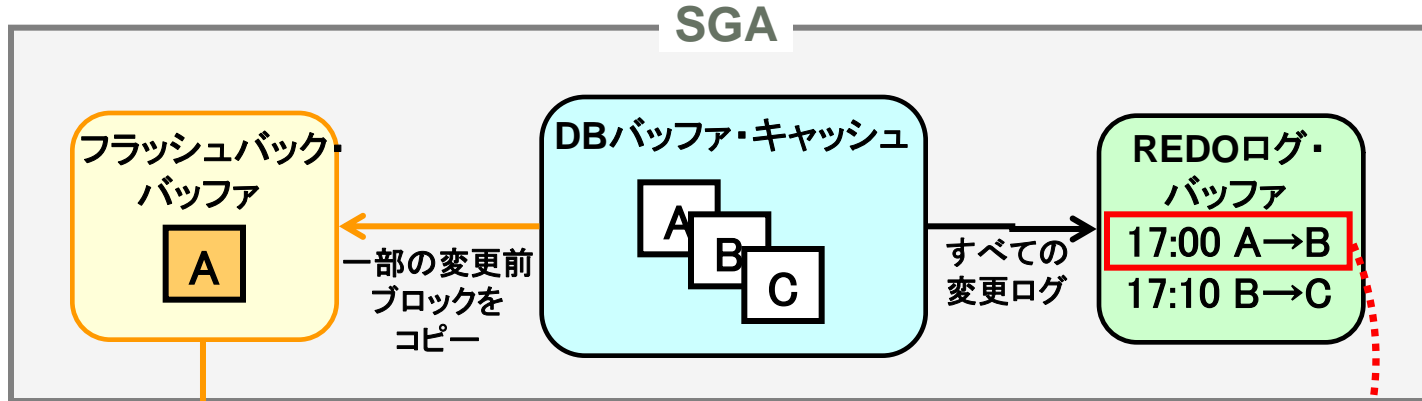
- フラッシュバック・データベースの有効化
  - フラッシュバック・データベース・ログを残す設定

## MOUNT状態から有効化

```
SQL> ALTER DATABASE FLASHBACK ON;  
  
SQL> SELECT flashback_on FROM v$database;  
FLASHBACK_ON  
-----  
YES
```

フラッシュバック・  
データベースの有効化

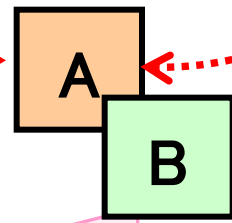
# フラッシュバック・データベースのしくみ



RVWR



17:05の時点でフラッシュバック



A→B

RVWRプロセスが、データ・ブロックに対する変更を非同期に取得 (ブロック単位でのバックアップ)

オーバーヘッドを抑えるためチェックポイントやコミットとは非同期で各ブロック15分に一度程度取得

フラッシュバック・データベース・ログとREDOログを使って、指定した時間までデータベースを戻す



# フラッシュバック・データベース 実行例

10g~

- フラッシュバック・データベース 実行例

時間やSCN指定が可能

```
SQL> FLASHBACK DATABASE TO TIMESTAMP (SYSDATE-1/24) ;
```

```
SQL> ALTER DATABASE OPEN READ ONLY ;
```

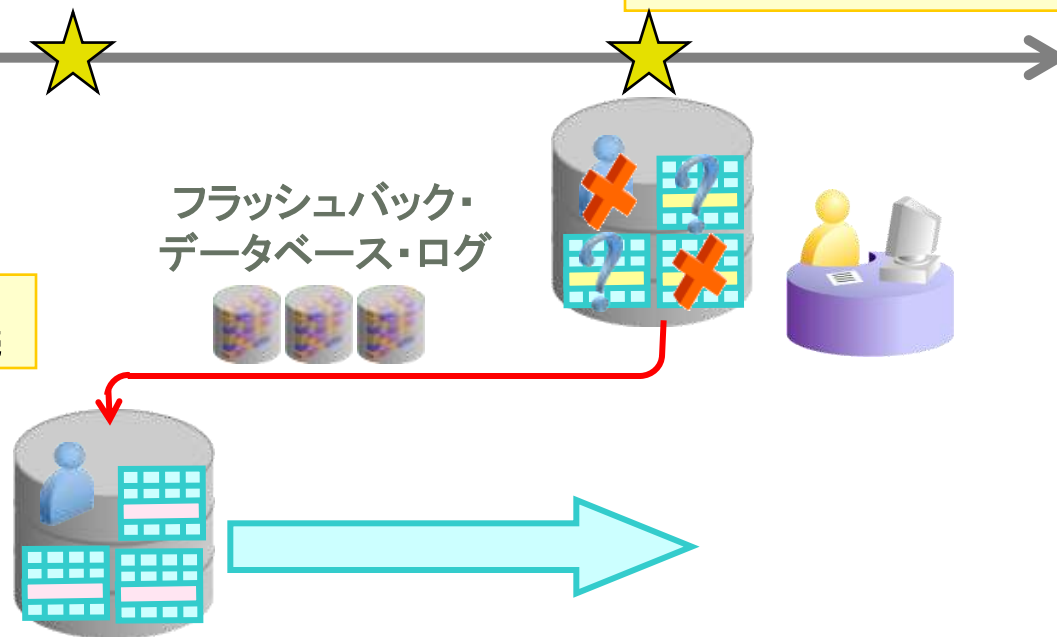
フラッシュバックを確定する前に  
READ ONLYでオープンして  
状態を確認することが可能

```
SQL> ALTER DATABASE OPEN RESETLOGS ;
```

最終的には「RESETLOGS」で  
データベースをOPEN

REDOログをリセットして  
新しいバージョンとして処理を継続

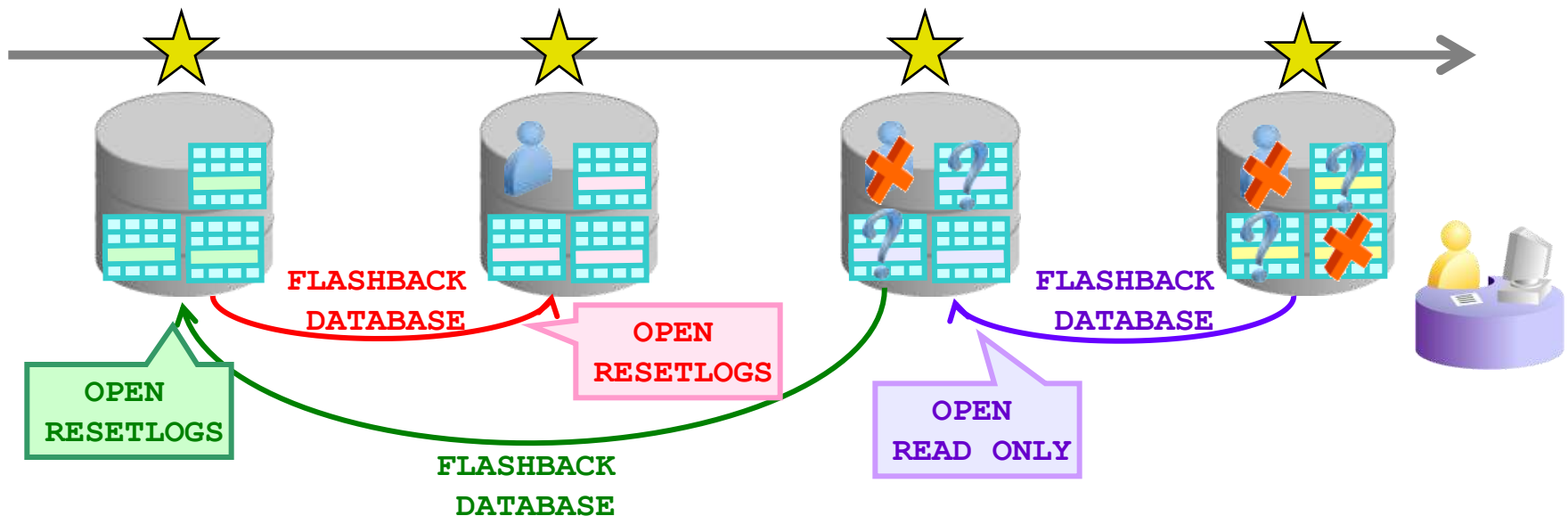
フラッシュバック・  
データベース・ログ



# フラッシュバック・データベースの特徴

10g~

- 何度でもやり直すことが可能なので、戻りたいデータポイントまで確実に戻ることができる
- 過去データの参照だけ行うことも可能なので、OPEN READ ONLY状態でデータをエクスポートし、フラッシュバック前に戻ることもできる
- RESETLOGSオープン後もフラッシュバックのやり直しが可能(10gR2~)



ORACLE

# まとめ

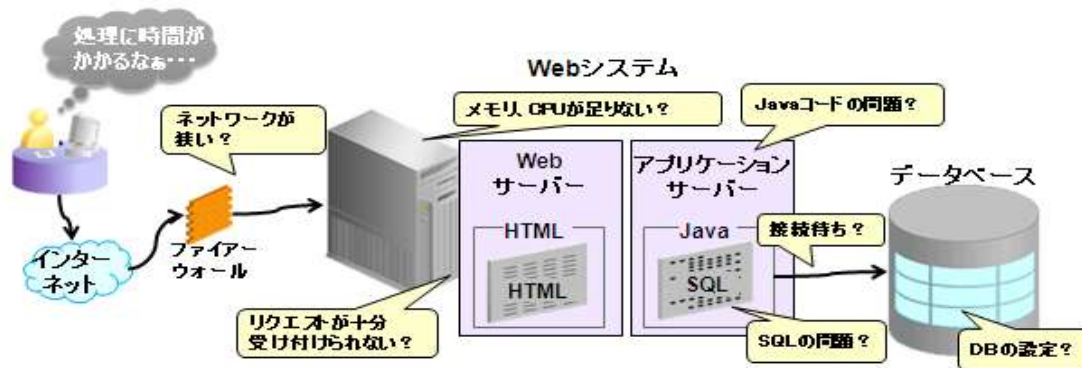
## 状況によって様々なフラッシュバック機能を使い分け 論理障害からの迅速な復旧を！

種類	機能名	使用場面	使用するログ	バージョン
参照	フラッシュバック問合せ	過去のデータの問合せ	UNDO情報	9i
	フラッシュバック・バージョン問合せ	どういう経緯で現在の値に至ったのかトランザクションの確認	UNDO情報	10g
	フラッシュバック・トランザクション問合せ	特定のトランザクションの追跡+ UNDO SQLの取得	UNDO情報	10g
	フラッシュバック・データ・アーカイブ	特定の表に対する長期間の履歴保存	フラッシュバック・アーカイブ領域	11g
リカバリ	フラッシュバック・トランザクション	誤って実行したトランザクションの取り消し	REDO情報	11g
	フラッシュバック・テーブル	誤って実行したDML文の取り消し	UNDO情報	10g
	フラッシュバック・ドロップ	誤って実行したDROPの取り消し	もとの表の領域を残す	10g
	フラッシュバック・データベース	大量の誤操作 表のTRUNCATE ユーザーの削除	フラッシュバック・データベース・ログ	10g

# Oracle Direct 新サービスができました

## 新規Oracle Direct Concierge (無償支援サービス)

- Webシステム ボトルネック診断サービス
- WebLogic Serverバージョンアップ支援サービス
- Oracle Application Server、その他アプリケーションサーバーから  
Oracle WebLogic Server への移行支援サービス



Oracle Direct 0120-155-096 

お問い合わせフォーム  
[http://www.oracle.co.jp/inq\\_pl/INQUIRY/quest?rid=28](http://www.oracle.co.jp/inq_pl/INQUIRY/quest?rid=28)

ORACLE

# OTN×ダイセミ でスキルアップ!!



- ・一般的な技術問題解決方法などを知りたい!
- ・セミナー資料など技術コンテンツがほしい!

Oracle Technology Network (OTN) を御活用下さい。

<http://otn.oracle.co.jp/forum/index.jspa?categoryID=2>

一般的技術問題解決にはOTN揭示版の  
「データベース一般」をご活用ください

※OTN揭示版は、基本的にOracleユーザー有志からの回答となるため100%回答があるとは限りません。  
ただ、過去の履歴を見ると、質問の大多数に関してなんらかの回答が書き込まれております。

<http://www.oracle.com/technology/global/jp/ondemand/otn-seminar/index.html>

過去のセミナー資料、動画コンテンツはOTNの  
「OTNセミナー オンデマンド コンテンツ」へ

※ダイセミ事務局にダイセミ資料を請求頂いても、お受けできない場合がございますので予めご了承ください。  
ダイセミ資料はOTNコンテンツ オン デマンドか、セミナー実施時間内にダウンロード頂くようお願い致します。

ORACLE

# OTNセミナー オンデマンド コンテンツ

期間限定にて、ダイセミの人気セミナーを動画配信中！！

ダイセミのライブ感はそのままに、お好きな時間で受講頂けます。

## 最新のコンテンツ



エンジニアのための  
ITIL実践術  
再生時間: 60分



ここからはじめよう  
Oracle PL/SQL入門  
再生時間: 60分



実践!!高可用システム構築  
-RAC基本  
再生時間: 60分



お悩み解決! Oracle  
のサイジング  
再生時間: 60分

## Database



今さら聞けない!!バックアップ-リカバリ入  
再生時間: 60分



意外と簡単!? Oracle Database 11g -セ  
再生時間: 60分



実践!!バックアップ-リカバリ  
再生時間: 60分



意外と簡単!? Oracle Database 11g -デ  
再生時間: 60分

>> もっと見る

## OTN オンデマンド

検索

※掲載のコンテンツ内容は予告なく変更になる可能性があります。  
期間限定での配信コンテンツも含まれております。お早めにダウンロード頂くことをお勧めいたします。

ORACLE

# オラクル クルクルキャンペーン

あのOracle Database Enterprise Editionが超おトク!!

おトクな買い方  
**オラクル5年分**

- ライセンス使用期間 を5年間に設定
- 初期のライセンスコストがなんと**67%OFF** !
- テクニカル・サポート価格も**53%OFF** !

**Enterprise Edition**はここが違う!!

- 圧倒的な**パフォーマンス!**
- データベース**管理がカンタン!**
- データベースを**止めなくていい!**
- もちろん**障害対策**も万全!

Oracle Databaseの  
ライセンス価格を**大幅に抑えて**  
ご購入いただけます

- 多くのお客様でサーバー使用期間とされる  
5年間にライセンス期間を限定
- 期間途中で永久ライセンスへ差額移行
  - 5年後に新規ライセンスを購入し継続利用
  - 5年後に新システムへデータを移行

この部分を  
お支払い

**67%  
OFF** ※2

Oracle Database

この機能でこの価格  
**ライセンスパック**

- Oracle Databaseの機能を**存分に使える!**
- **2ノードRAC**構成も可能!
- サーバー構成によって計**4種類**のパックから**選べる!**

詳しくはコチラ<http://www.oracle.co.jp/campaign/kurukuru/index.html>

Oracle Direct 0120-155-096 

お問い合わせフォーム

[http://www.oracle.co.jp/inq\\_pl/INQUIRY/quest?rid=28](http://www.oracle.co.jp/inq_pl/INQUIRY/quest?rid=28)

ORACLE



あなたにいちばん近いオラクル



# Oracle Direct

まずはお問合せください

Oracle Direct

検索

システムの検討・構築から運用まで、ITプロジェクト全般の相談窓口としてご支援いたします。  
システム構成やライセンス/購入方法などお気軽にお問い合わせ下さい。

Web問い合わせフォーム

専用お問い合わせフォームにてご相談内容を承ります。

[http://www.oracle.co.jp/inq\\_pl/INQUIRY/quest?rid=28](http://www.oracle.co.jp/inq_pl/INQUIRY/quest?rid=28)

※フォームの入力には、Oracle Direct Seminar申込時と同じ  
ログインが必要となります。

※こちらから詳細確認のお電話を差し上げる場合がありますので、ご登録されている連絡先が最新のものになっているか、ご確認下さい。

フリーダイヤル

0120-155-096

※月曜~金曜 9:00~12:00、13:00~18:00

(祝日および年末年始除く)

ORACLE





以上の事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。以下の事項は、マテリアルやコード、機能を提供することをコミットメント(確約)するものではないため、購買決定を行う際の判断材料になさらないで下さい。オラクル製品に関して記載されている機能の開発、リリースおよび時期については、弊社の裁量により決定されます。

Oracle、PeopleSoft、JD Edwards、及びSiebellは、米国オラクル・コーポレーション及びその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標の可能性がります。