

# 障害時回復

# 内容

- 障害の種類
- 障害回復手法
  - ログ法
  - シャドウページ法
- ログ法
  - WAL
  - チェックポイント

# 障害時回復

- データベースの一貫性を保証するための必須機能
- 障害が発生しても,
  - コミットしたトランザクションが行った操作を残さなければならない
  - コミットしていないトランザクションが行った操作はなかったことに
- データベースの一貫性を損なう障害とは
  - **トランザクション障害**: 誤入力, データが見つからない, 計算エラー, ...
  - **システム障害**: データベースシステムやOSの障害, ハードウェア障害, 電源断, ...
  - **メディア障害**: ディスククラッシュ, ...

# トランザクション単位の回復

トランザクションの**原子性や耐久性の保証**のための回復機能

- 障害発生時に実行中 (コミット前) のトランザクションが行った操作を、**一切なかったことにする (UNDO処理)**
- 障害発生時に**コミットされていたが**, 2次記憶 (データベース) まで反映されていない**書き込みの結果を**, **回復時に2次記憶に反映する (REDO処理)**

障害の種類 回復の手順	トランザクション障害	システム障害	メディア障害
UNDO	トランザクションUNDO	全局的UNDO	—
REDO	—	局所的REDO	全局的REDO

全局的 (global): すべて    局所的 (local): 当該必要な

# 障害回復手法

- ログ法 (12.2節)
  - データベースへの操作をログとして不揮発性の媒体に記録しておき、障害時にはそのログを用いてREDOやUNDOを行う
  - ログは、トランザクションの開始、終了、トランザクションが行ったデータベースへの読み出し、書き込み、チェックポイント情報などの時系列データ (p.141, 図12.2参照)
- シャドウページ法 (コラム (p. 148))
  - コミット前の領域をシャドウページとして保存しておき、データベースへの更新を上書きでなく、別の領域 (ページ) で行い、コミット時に新しいページを正式なページとする (p.148, 図12.6参照)  
UNDO時には、シャドウページを正式なページに戻す
  - REDO, UNDOを必要としない  
(ただし、コミット前に2次記憶への確実な反映が必要)

# ログ法でのWALプロトコル

- データベースを**更新する前にログに操作内容を記録する** (**write **a**head log**)
  - **ログファイルへの書き込みは2次記憶に確実に反映させる** (強制フラッシュ)
  - **トランザクションの書き込みはOSに任せる** (即時更新) が、場合によっては**コミット操作がログに書き込まれるまで、2次記憶への書き込みを許さない** (遅延更新) モードもある

**データベースはコミットされたトランザクションの操作のみ反映される**

# チェックポイント法

- 適当な時点 (チェックポイント) でバッファ (メモリ中) の内容をデータベース (2次記憶) に強制書き出しを行うことで、**ログの内容をデータベースの内容と同期させる**

これにより、REDO, UNDOの時間短縮および無駄なREDOを削減する

- 次の操作を行う
  1. 実行中のトランザクションを全て一時停止する
  2. **バッファの内容をデータベースに強制書き出し**
  3. **チェックポイントレコードをログに書き出す**
  4. 中断させていたトランザクションを再開する

# チェックポイント法での回復処理

- データベースの状態をチェックポイント時点に戻す
- チェックポイント以前にコミットされていないトランザクションを障害時の状態に応じてREDOあるいはUNDOを行う
  - 障害時にコミットしていなかった場合  
UNDOする
  - 障害前にコミットしていた場合  
REDOする



