

- 共有ロック ✕ 共有ロック ●
→ どちらも参照のみで値が変わらないからできる
- 共有ロック ✕ 占有ロック ✕
→ 先に共有ロックをかけても、参照している値が更新されるためデータの一貫性（データがある規則によってのみ変更されないこと）がなくなる
- データの一貫性：ここでの終始変わらないとは特定の規則以外ではデータの変更などが起きないこと

障害管理

主記憶

DBで更新を行うと、更新したログが出力されますか？

チェックポイント：主記憶上にあるデータが

実際に SQL 文で処理されたデータは DB サーバーの主記憶上で変わっているだけで何らかのタイミングで DB などのストレージに保存されるらしいですが、それはいつですか？コミットした時などでしょうか？

DBMS でデータが DB サーバーのディスクに保存されるまでの流れ

1. SQL 文を実行。

- DB サーバーのキャッシュやメモリ上でテーブルで取得した値を変更する
- この変更はトランザクションログに記録される
 - ◆ **トランザクションログ (Transaction Log)**
 - ◆ **記録タイミング:** SQL 文が実行された時点で、その操作内容（例えば、INSERT、UPDATE、DELETE）の詳細がログに記録されます。
 - ◆ **内容:** 更新対象のデータの変更前の値と変更後の値。
 - ◆ 操作の種類（例: UPDATE、INSERT）。
 - ◆ 操作されたテーブルや列の情報。
 - ◆ **目的:** データ復旧（リカバリー）や障害発生時のロールバックを可能にするため。
 - ◆ **例:** MySQL の binary log や PostgreSQL の WAL (Write-Ahead Logging) が該当します。

2. コミット

- トランザクションが正常に終了するとコミットが行われ、完全にデータがディスクに書き込まれ、永続になる
- 書き込むデータ元は 1 のメモリ上に格納したデータです

3. チェックポイント

- メモリで格納したデータをストレージに書き込むタイミングのこと
- ストレージにデータを書き込む速度が遅いので、一定のタイミングで書き込むようになっている

トランザクションログと障害管理に対しての基本的な捉え方はこれでよろしいでしょうか？

- SQLを実行したら、処理の成否に問わずトランザクションログはファイルとして記録される
 - コミットが成功しているものはチェックポイントのタイミングでログの内容を元にDBのストレージに変更内容が書き込まれ、ログも永続的に残るように保存される
 - 未コミットの場合はログの更新前の情報を元にSQL実行前の状態に戻り、ログも破棄される（後述のような復旧対策として残す可能性もある）
- コミットが成功した場合のトランザクションログは、主に障害復旧時のロールフォワードに利用され、未コミットのトランザクションログは障害復旧時のロールバックとSQL実行時の失敗の際のロールバックに利用する
- トランザクションログはSQL実行時の更新前後の値のログと言い換えても問題ない
- DB障害を受けてサーバー自体の電源が落ちるとメモリ上のデータが失われるが、トランザクションログはディスク上に格納されているので、それを元にDBの状態を復元する
 - 上記方法は標準的な障害復旧方法であり、ポイント・イン・タイム・リカバリ（PITR: Point-In-Time Recovery）と呼ばれる