【ACID特性を極める】

基本事項

ACID特性は、トランザクションが持つべき性質ですが、ACID特性を説明するにあたり、多くの用語が出てくるため、まず、これらの用語を以下のとおりまとめました。

トランザクション（XACT）

①**XACT**は、一連の処理の単位であり、**XACT**の完了を**コミット**と呼ぶ。**コミット**情報を**ログファイル**に書き出した時点で**コミット**が完了して、**XACT**も完了する。
②**XACT**内の全ての処理が成功し、全ての**ログ**が**ログファイル**に書き込まれ、その後の直近の**チェックポイント**により、全ての処理がＤＢに反映される。
③よって、 **XACT**完了までロックを掛けると、他の**XACT**は中途半端に更新されたＤＢを参照する心配はない。

ログファイル

①メモリ上のログバッファ領域にある**ログ**を記憶装置の**ログファイル**に書き出す。
②ログファイルは、DBに対する操作履歴を記録するファイルであり、記憶装置に保存されている。
③ログファイルの中身は、処理の種類、更新時刻、**ログ（更新前イメージと更新後イメージのペア）**などである。

ログと障害回復

① DBの利用は、利用するデータを記憶装置からメモリ上に読み込む。そして、メモリ上のデータを更新する場合には、個々のデータを更新する毎に**メモリ上のログバッファ領域**に**ログ**が生成される。以上はメモリ上の処理であり、更新されたデータや**ログ**の記憶装置への書き出しのタイミングは下記②と③のとおりである。
②**ログ**の**ログファイル**への書き出しのタイミングは、**XACT**の**コミット時点**である。
③データの記憶装置への書き出しのタイミングは、**チェックポイント**が来た時であり、②とは全く関係なく実行される。つまり、**XACT**中の中途半端にデータが更新された状態であっても、記憶装置に書き出される。
④**XACT**中に**チェックポイント**があり、記憶装置のデータが中途半端に更新された状態において、システム障害により**XACT**が中断しても、**ログファイル**にある**ログ**の更新前イメージによって、記憶装置のデータを**XACT**が全く実行されなかった状態に復旧できる。これを**ロールバック（UNDO：取り消し）**という。
⑤上記④の**ロールバック**を可能とするためには、更新前イメージと更新後イメージを比較して、データが更新されているが否かを判別する必要がある。たからこそ、更新前イメージと更新後イメージのペアが**ログ**なのである。よって、チェックポイントの前に**ログ**が**ログファイル**に書き出されていなければならず、これを**WAL（Write Ahead Logging：ログ先書き）**という。当然、次の⑥の時にも**WAL**は必須である。
⑥**XACT**完了後から、その直後の**チェックポイント**までの間にシステム障害等があれば、**ログファイル**にある**コミット**後の完全な更新後イメージから、記憶装置のデータを**XACT**が完了した最新の状態にすることができる。これを**ロールフォワード（REDO：再実行）**という。

隔離性水準

ACID特性を説明する前に、もう１つ理解すべきこととして、隔離性水準（分離レベル）があります。**参照側及び更新側トランザクション（XACT）中のロック状態**により、下表のとおり分類されます。

隔離性水準（分離レベル）	参照側及び更新側のロックの状態	独立性阻害要因
READ UNCOMMITED	**ロック粒度は行**とする。 ●更新側XACTは、更新時に専有ロックを掛け、XACT終了まで解放しない。 ●参照側XACTは、共有ロックは掛けない。 ●参照側は、更新側の専有ロックを無視して、参照できるらしい。	●更新側の更新途中（コミット前）のデータを参照できてしまうため、参照後にロールバックされて存在しなくなった当該データを使い続ける**「DIRTY READ (DR)」**が発生する。 ●**DR**, **NR**, **FR**全てが発生する。
READ COMMITED	**ロック粒度は行**とする。 ●参照側XACTは、参照時に共有ロックを掛け、XACT中であっても、参照終了時に解放する。 ●更新側XACTは、更新時に専有ロックを掛け、XACT終了まで解放しない。 ●更新側XACTがロック中は参照側は、更新途中のデータは参照できない。	●参照側XACT中に、更新側XACTによる対象行の更新の確定（コミット）により、参照側XACT中に、対象行を参照するたびに内容が変わる**「NONREPEATABLE READ (NR)」**が発生する。 ●**DR**は発生しないが、**NR**, **FR**が発生する。 **注）NONREPEATABLE READ (NR)は、UNREPEATABLE READ（UR）でもOK**
REPEATBLE READ	**ロック粒度は行**とする。 ●参照側XACTは、参照時に共有ロックを掛け、XACT終了まで解放しない。 ●更新側XACTは、更新時に専有ロックを掛け、XACT終了まで解放しない。 ●参照側が「SELECT ＊ FROM A」で参照した行に対する共有ロックは、XACT中は解放されないため、更新側は更新できない。	●更新側が新たに追加する行は、参照側による共有ロックの対象外のため、参照側が再度「SELECT ＊ FROM A」を実行した場合、その追加した行が出現する**「FANTOM READ (FR)」**が発生する。 ●**DR**, **NR**は発生しないが、**FR**が発生する。
SERIALIZABLE (直列化可能性)	●参照側は、参照時からXACT終了までの共有ロックの**粒度をテーブル**にする。 ●更新側は、テーブル全体に対する参照側からのロックにより、全ての行に対する追加、更新及び削除はできなくなる。	●**直列化可能性**（XACTを複数並行で実行しても、完全独立して順番に実行しても、結果は同じこと）は確保されるが、排他制御によりスループットは格段に悪化。 ●**DR**, **NR**, **FR**は発生しない

ACID特性のまとめ

以上の説明によって、ロールバック、ロールフォワード、コミット、隔離性水準等について、理解が深まったので、ACID特性を下表（ACID特性）のとおり、まとめました。

表：ACID特性

ACID特性	左記の意味	実現する機能
原子性 (Atomicity)	トランザクションは完全に実行されるか、全くされないかのどちらかである。	**コミットメント制御**として、**コミット**（トランザクション完了によるデータ変更の確定）と**ロールバック**（トランザクションがなかったことにすること）
一貫性 (Consistency)	トランザクションの実行前と実行後でデータの整合性がある。	**ロック**による**排他制御（同時実行制御）**として**直列化可能性** ※下記の**隔離性水準**を**SERIALIZABLE**にすると**直列化可能性**
独立性 (Isolation)	複数のトランザクションは独立して実行できる。	**ロック**による**排他制御（同時実行制御）**として**隔離性水準** ※完全な独立性を満たすにはの**隔離性水準**を**SERIALIZABLE**にすることであるが、スループットが大きく低下するため、それと信頼性のバランスを考えて**隔離性水準**を決める。
耐久性 (Durability)	トランザクションの結果（コミット後のデータ）は、障害が発生しても回復できる。	**障害回復**として、**ロールバック**（更新前イメージによるDBの回復）と**ロールフォワード**（更新後イメージによるDBの回復） ※今あるDBは、過去の全てのトランザクションの結果であるという意味と解釈すれば、**ロールバック**はそれを回復させる。

ロストアップデート

前ページの表（隔離性水準）は、一方が参照側、他方が更新側によるものでしたが、両方とも更新側である場合、排他制御（ロック）を全く行わないと、以下の例のとおり、**更新データの喪失（ロストアップデート）**が発生します。

①両方ともロックなしで、10というデータをメモリ上に読み込む。
②一方が10に10を加算して20となったメモリ上のデータは、記憶装置に書き出される。
③他方が10に10を加算して20となったメモリ上のデータは、記憶装置に書き出される。
④本来ならは、10に対して一方と他方がそれぞれ10加算するので30となるはずが、記憶装置には20が書き出される。

よって、一方がデータをメモリ上に読み込む前にロック**（共有ではなく専有ロック）**することで、他方のデータの読み込みさせなければ**ロストアップデート**は発生しません。