バックアップとリストア、リカバリ

# 信頼性の為に備えるべき４つの特性

ITの世界では、「**データを正確かつ安全に取り扱うためにシステムが備えるべき４つの特性**」として、**ACID特性**というものが広く知られています。データベースでは情報を管理するために利用されるので、DBMSはACID特性を満たすことが求められます。

・ACID特性とは

原子性

Atomicity

永続性

Durability

分離性

Isolation

一貫性

Consistency

年齢

-256

update

update

|  |  |  |  |
| --- | --- | --- | --- |
| 特性 | | 説明 | DB機能 |
| A | 原子性 | 処理が中断しても中途半端な更新をしない | コミット、ロールバック |
| C | 一貫性 | データの内容が矛盾した状態にならない | データ型、制約 |
| I | 分離性 | 処理を同時実行しても副作用がない | ロック、読取り一貫性 |
| D | 永続性 | **記録した情報は消滅せずに保持される** | **バックアップ、リストア** |

データベースに格納された情報は、ハードディスクなどの磁気記憶媒体に記録されます。

しかし、ハードディスクは物理的な存在である以上、突然データの読み書きが出来なくなる可能性もゼロではありません。そのような事態が発生しても、**情報の永続性を確保するためバックアップとリストアの仕組みがDBMSには備わっています**。

# バックアップとリストアの違い

バックアップとは、障害が発生した時にデータベースの状態を復元できるように

あらかじめに必要なデータなどを取得して保存することです。

リストアとは、障害が発生した時にデータベースの状態を復元する仕組みや行為の事です。

バックアップ：

障害が発生した時にDBを復元できるように

DBのデータを別途保存すること

リストア：

障害が発生した時にバックアップを使用して

DBの状態を元に戻すこと

4/1

4/2

4/3

4/5

4/4

バックアップ

リストア

障害発生

# オンラインバックアップとオフラインバックアップ

もし、INSERT文やCREATE TABLE文などでデータベースの内容を書き替えている間にバックアップが行われると、バックアップファイルは中途半端な状態となり、バックアップデータとして整合性が取れなくなる恐れがあります。

**整合性を保ちつつバックアップを行う最も簡単な方法は、データベースを停止してからバックアップを取るオフラインバックアップ**です。オフラインバックアップのデメリットは

バックアップ中はデータ処理が一切行えないことです。そこで多くのDBMSは、**データベースを稼働しながら整合性のあるバックアップデータを取得できるオンラインバックアップ**があります。

・バックアップの種類

|  |  |  |
| --- | --- | --- |
| 種類 | 説明 | メリット・デメリット |
| オフライン  バックアップ | DBを停止させて  バックアップを取得 | 簡単にバックアップが取得できる反面  クライアントはDBが利用できない。 |
| オンライン  バックアップ | DBを稼働したまま  バックアップを取得 | クライアントはバックアップを意識せずに  DBの利用可能。しかしバックアップ方法はDBMSに違いが出てくる。 |

# 論理バックアップと物理バックアップ

バックアップを取得する方法には、論理バックアップと物理バックアップがあります。

物理バックアップは、データベースのデータファイルをコピーして取得する方法です。

簡単に実施できますが、オフラインバックアップにしか対応していないのであまり利用しません。DBMSのバックアップツールを利用してバックアップを取得する手法を論理バックアップと言います。データベースを起動しながら実施できるので基本的にバックアップを取得する時には論理バックアップを使用します。

MySQL

データ

ファイル

MySQL

データ

ファイル

バックアップ

ファイル

データ

ファイル

・論理バックアップ

・物理バックアップ

バックアップ

バックアップ

# mysqldumpコマンドによる論理フルバックアップ

mysqldumpとはMySQLの論理フルバックアップ取得ツールです。インスタンス全体、

データベースごと、テーブルごとなど必要な単位でバックアップを作成できます。

・mysqldumpコマンド基本書式

|  |
| --- |
| 全てのデータベースを対象としたバックアップ |
| mysqldump -u ユーザ名 -p --all-databases > バックアップファイル名 |
| 複数のデータベースを対象としたバックアップ |
| mysqldump -u ユーザ名 -p --databases DB名1 DB名2 > バックアップファイル名 |
| 特定のデータベースのみ対象としたバックアップ |
| mysqldump -u ユーザ名 -p データベース名 > バックアップファイル名 |
| 特定のテーブルのみ対象としたバックアップ |
| mysqldump -u ユーザ名 -p データベース名 テーブル名 > バックアップファイル名 |

mysqldumpには、オプションを指定できます。主なオプションは以下の通りです。

・mysqldumpコマンドの主なオプション

|  |  |
| --- | --- |
| オプション | 説明 |
| --host | バックアップ対象のMySQLサーバーのホスト名 |
| --port | バックアップ対象のMySQLサーバーのポート番号 |
| --flush-logs | バックアップと共にログを切り替える |
| --master-data | バイナリログファイル名とポジションを出力する |
| --single-transaction | バックアップを１つのトランザクション内で実行する |
| --no-data | データを取得せずに、DBやテーブルの定義のみ取得する |

ハンズオン　バックアップを作成する

１．コマンドプロンプトを開く（mysqlにログインしないでください。）

２．mysqldumpコマンドでstudbのバックアップを取得する

mysqldump -u root -p studb --single-transaction > studb\_bk1.sql

３．rootパスワードを入力して、バックアップ処理を実行する。

４．作成されたstudb\_bk1.sqlをテキストエディタで開いて中身を確認する。

single-transactionオプションを付けることにより、バックアップ処理が一つのトランザクション内で行われる為、バックアップ実行中に他のユーザが不用意にデータを書き換えられるのを防ぐことが出来ます。しかし、ALTER TABLEなどのDDL文を実行するとデータの一貫性が保てなくなるのでバックアップ中は命令を実行しないように注意しましょう。

作成されたバックアップファイルを確認すると、CREATE TABLE文とINSERT文で構成されていることがわかると思います。mysqldumpの論理バックアップでは、データベースを構築するためのDDL文が作成され、リストアではこのファイルを順に実行してくことでデータベースの復旧を図ります。

# バックアップのリストア

バックアップファイルをデータベースに反映する行為をリストアと言います。

MySQLでは、mysqlコマンドを使用してバックアップファイルの読み込みを行います。

・リストアのコマンド

mysql -u root -p < バックアップファイル名

ハンズオン　データベースをリストアする

１．mysqlにログインして、bikeテーブルの削除を行いログアウトする。

mysql -u dbuser -p studb

DROP TABLE BIKE;

EXIT;

２．rootユーザでバックアップファイルを読込みデータベースのリストアを行う

mysql -u root -p studb < studb\_bk1.sql

３．再度、dbuserでMySQLにログインしてbikeテーブルを確認する

mysql -u dbuser -p studb

SELECT \* FROM BIKE;

このように、バックアップとリストアを行うことでデータベースに障害が発生しても復元することができます。しかしそれだけだとまだ、データベースを再開することが出来ません。

4/1

4/2

4/3

4/5

4/4

バックアップ

リストア

障害発生

**リストアだけだと4/1の状態に戻っただけ、**

**4/2~4/5のデータは復旧出来ていない。**

# バイナリログと増分バックアップ

mysqldumpを使って行うような、データベース全体のバックアップのことをフルバックアップと呼びます。それに対して、バックアップを取得した以降の変更点のみを対象としたバックアップを増分(差分)バックアップと呼びます。

フルバックアップはデータ全量が対象になるため、バックアップのためにたくさんの時間とディスク容量が必要になりますが、増分だけであれば小さく済みます。そのため増分バックアップは頻繁に取得しやすいです。MySQLではバイナリログを利用して実現することが出来ます。

4/1

4/2

4/3

4/5

4/4

バックアップ

ログ

ログ

ログ

INSERT INTO …

UPDATE …

INSERT INTO …

バイナリログはMySQLに対する変更イベントを記録するので、以前と同じ再現を行うことが出来ます。つまり、データベースの復旧を行うときにログに記載されている処理を再実行することで、障害が発生する直前の状態になるという仕組みです。

バイナリログに関する設定は、環境設定ファイルで指定を行いますが

MySQL8.0ではデフォルトで有効化されています。

・バイナリログ関連システムパラメータ

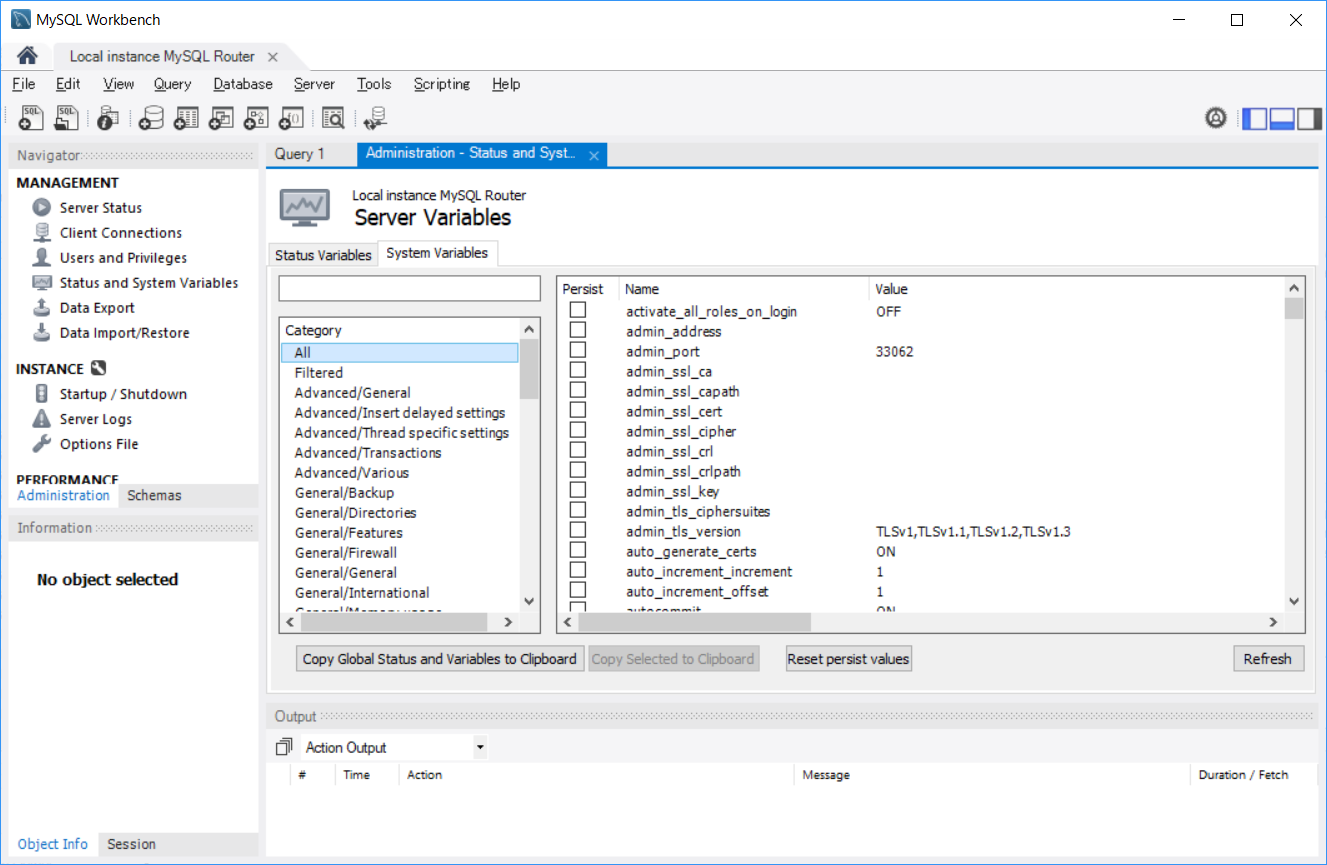
|  |  |
| --- | --- |
| システムパラメータ | 説明 |
| datadir | データディレクトの場所。  データファイルやバイナリログなどが管理されている。 |
| log\_bin | バイナリログの出力設定。MySQL8.0はデフォルトでON |
| log\_bin\_basename | バイナリログのベースとなる名前 |
| binlog\_format | バイナリログの出力形式。MySQL8.0のデフォルトはROW  STATEMENTとMIXEDもあるが、基本的にROWでOK |
| server\_id | MySQLのサーバーを識別するID。デフォルトは１。  複数台で構成される場合は、重複しない値を設定する。 |

ハンズオン　バイナリログの設定を確認する

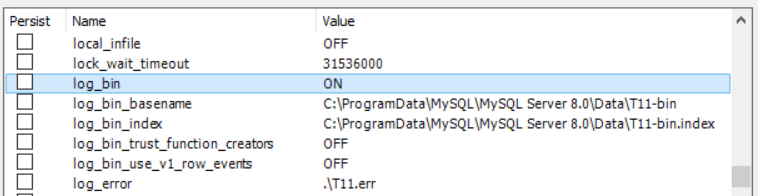
１．MySQL Workbenchを起動させて、rootユーザでログインを行う。

２．「Status and System Variables」をクリックして、「System Variables」タブを選択

Categoryには「All」を選択する。



３．「log\_bin」と「log\_bin\_basename」のパラメータ値を確認する。



４．指定されているディレクトリに移動して、バイナリログが存在するか確認する。

**※必ず、自分の調べたパラメータ値でバイナリログの存在チェックをすること！！**

# ロールフォワードリカバリ

ロールフォワードリカバリとは、フルバックアップと増分バックアップを併用することで障害発生直前、または特定のポイントまでのデータを復活させる手法です。

MySQLでは、mysqlbinlogコマンドでバイナリログに記録された変更イベントを取得してmysqlコマンドで再生することが出来ます。

・ロールフォワードリカバリ

4/1

4/2

4/3

4/5

4/4

バックアップ

ログ

ログ

ログ

リストア

4/1分反映

4/2分反映

4/3分反映

4/4分反映

ロールフォワード

リカバリ

ログに記録されているSQL文を再実行して、障害が発生する直前の状態までデータを更新する処理のことを**ロールフォワード**(roll forward)と言います。ロールバックと名前が似ていますが、処理内容は対照的なので混同しないようにして下さい。なお、ロールフォワードを行いデータベースが障害直前まで復旧することを**リカバリ**と言います。

|  |  |
| --- | --- |
| 処理名 | 処理の違い |
| ロールバック | 実行した処理を取り消す。  SQLの実行失敗やデッドロックなどで利用される。 |
| ロールフォワード | 障害復旧時に、バックアップからリストアした後に  バイナリログを使用して障害直前までの処理を再実行する。 |

ハンズオン　ロールフォワードリカバリを実施する。

１．全体バックアップを取得する。

mysqldump -u root -p studb --flush-logs --single-transaction --master-data > studb\_bk2.sql

２．バックアップファイルをテキストで開いて、MASTER\_LOG\_FILEのファイル名を

確認する。

３．MASTER\_LOG\_FILEで指定されているバイナリログが新しく作成されていることを

確認する。

４．MySQLにdbuserでログインして、studbデータベースの変更を行う。

INSERT INTO TAX (TAX, START\_DATE, END\_DATE, TAKEOUT\_FLG)

VALUES (5, '1997-04-01', '2014-03-31', 0);

INSERT INTO TAX (TAX, START\_DATE, END\_DATE, TAKEOUT\_FLG)

VALUES (8, '2014-04-01', '2019-09-30', 0);

INSERT INTO TAX (TAX, START\_DATE, END\_DATE, TAKEOUT\_FLG)

VALUES (10, '2019-10-01', NULL, 0);

INSERT INTO TAX (TAX, START\_DATE, END\_DATE, TAKEOUT\_FLG)

VALUES ( 8, '2019-10-01', NULL, 1);

COMMIT;

５．MySQLからログアウトする。

EXIT;

６．バイナリログが更新されているか確認する。

mysqlbinlog --verbose "ここは各自のバイナリログの名前！！"

**ここから、疑似的にデータベースを破壊します！！**

**注意して作業してください！！**

７．MySQLサーバーを停止してください。

８．バイナリログをdbwkフォルダにコピーしてください。

９．studbデータファイルの一部をデスクトップに移動します。

C:\ProgramData\MySQL\MySQL Server 8.0\Data\studb内にある

product.ibdファイルをデスクトップに移動してください。

１０．MySQLサーバーを起動してください。

１１．studbに接続して、product表の取得でエラーを確認してください。

SELECT \* FROM PRODUCT ;

ここからリストアおよびロールフォワードリカバリを開始します。

１２．バックアップファイルからデータベースのリストアを行う。

mysql -u root -p studb < studb\_bk2.sql

１３．studbに接続して、product表とtax表のデータを取得してください。

SELECT \* FROM PRODUCT ;

SELECT \* FROM TAX;

EXIT;

１４．バイナリログを使ったロールフォワードを行います

mysqlbinlog バイナリログ名 | mysql -u root -p

１５．studbに接続して、product表とtax表のデータを取得してください。

SELECT \* FROM PRODUCT ;

SELECT \* FROM TAX;

EXIT;

# バイナリログの削除

バイナリログは、MySQLを稼働していると増え続けます。サーバーのディスク容量に負荷をかけない為、デフォルトでは３０日間保存されます。binlog\_expire\_logs\_secondsの設定を変更することで任意の期間に変更できます。