

# MySQL

[Database][CentOS]

- ・ [Django インストール 1.5.1](#)
- ・ [CentOS 初期設定](#)

## インストール

### Windows

- ・ [Windows8.1 に MySQL5.6 をインストール](#)

### Ubuntu(apt-get でインストール)

```
$ sudo apt-get install mysql-server
```

- ・ 途中でパスワードの設定を求められる

### CentOS(yum からインストール)

- ・ <https://dev.mysql.com/doc/refman/5.6/ja/linux-installation-yum-repo.html>

### 手順

- ・ <https://dev.mysql.com/downloads/repo/yum/> に移動
- ・ 使用しているプラットフォーム用のリリースパッケージを選択してダウンロード
- ・ インストール

```
$ sudo yum localinstall mysql80-community-release-el7-3.noarch.rpm
yum update
$ sudo yum install mysql-community-server
```

### ダウンロード

- ・ <http://www.mysql.com/>
- ・ <http://dev.mysql.com/downloads/>

### MySQL Community Server

- ・ <http://dev.mysql.com/downloads/mysql/>
- ・ [MySQL-5.6.13-1.linux\\_glibc2.5.i386.rpm-bundle.tar](#)

### アカウントの登録

- ・ mysql グループと mysql ユーザーの作成

```
# groupadd mysql
# useradd -g mysql mysql
```

### 解凍

```
$ tar xvf MySQL-5.6.13-1.linux_glibc2.5.i386.rpm-bundle.tar
```

## インストール

- ・ 標準の最低限のインストールには、サーバーとクライアント RPM をインストールします。
- ・ RPM はまた MySQL サーバーを運用するためのユーザー `mysql` (存在しない場合) のログインアカウントを作成する
- ・ サーバーがブート時に自動的に起動するように適切なエントリを `/etc/init.d/` に作成する

```
# rpm -ivh MySQL-shared-5.6.13-1.linux_glibc2.5.i386.rpm
# rpm -ivh MySQL-shared-compat-5.6.13-1.linux_glibc2.5.i386.rpm
# rpm -ivh MySQL-server-5.6.13-1.linux_glibc2.5.i386.rpm
# rpm -ivh MySQL-client-5.6.13-1.linux_glibc2.5.i386.rpm
```

server インストール時に、初期パスワードの情報がコンソールに出力される

A random root password has been set. You will find it in '/root/.mysql\_secret'.

もしくはログに出力されている

```
/var/log/mysqld.log | grep "temporary password"
```

開発で必要 (Django など)

```
# rpm -ivh MySQL-devel-5.6.13-1.linux_glibc2.5.i386.rpm
```

## 起動と停止

### CentOS

```
# /sbin/service mysql start
Starting MySQL... [ OK ]
# /etc/init.d/mysql stop
Shutting down MySQL.. [ OK ]
```

### CentOS7

```
# systemctl enable mysqld
# systemctl start mysqld
```

### Ubuntu

```
$ sudo /etc/init.d/mysql start
$ sudo /etc/init.d/mysql stop
$ sudo /etc/init.d/mysql restart
```

mysqld\_safe ( mysqld のラッパ )

```
shell> cd mysql_installation_directory
shell> bin/mysqld_safe &
```

## 起動設定がされたか確認

```
# chkconfig | grep "mysql"
mysql          0:off  1:off  2:on   3:on   4:on   5:on   6:off
```

## 権限の初期設定

- <http://dev.mysql.com/doc/refman/5.1-olh/ja/unix-post-installation.html>

## 忘れたパスワードをリセット

- <http://dev.mysql.com/doc/refman/4.1/ja/resetting-permissions.html>

mysqld を --skip-grant-tables オプションで起動

```
# mysqld --skip-grant-tables &
```

mysqld サーバに接続

```
# mysql -u root mysql
```

## 新しいパスワードの設定

```
mysql> update user set Password=PASSWORD('newpassword')
-> where User='root';
Query OK, 4 rows affected (0.11 sec)
Rows matched: 4  Changed: 4  Warnings: 0

mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)
```

## root パスワードの変更

```
mysql> set password for 'root'@'localhost' = password('newpassword');
Query OK, 0 rows affected (0.00 sec)
```

## パスワードの設定 MySQL8

- [https://qiita.com/arm\\_band/items/12208908041a5506d7f4](https://qiita.com/arm_band/items/12208908041a5506d7f4)

```
mysql> set password for 'wordpress'@'localhost' = 'your password';
```

## 初期の権限変更ツール mysql\_secure\_installation

- 初期パスワード /var/log/mysqld.log | grep "temporary password"

```
# mysql_secure_installation
Change the root password? [Y/n] n
Remove anonymous users? [Y/n] Y
Disallow root login remotely? [Y/n] Y
Remove test database and access to it? [Y/n] n
```

Reload privilege tables now? [Y/n] Y

All done! If you've completed all of the above steps, your MySQL installation should now be secure.

Thanks for using MySQL!

Cleaning up...

## 簡易設定

### バージョンの確認

```
mysql> select version();
+-----+
| version() |
+-----+
| 5.0.77    |
+-----+
1 row in set (0.00 sec)
```

### MySQL 簡易設定

- ・ [MySQL 簡易設定](#)

## データベースの作成

### 作成

```
# mysql -u root -p
mysql> create database test_db default character set utf8;
Query OK, 1 row affected (0.00 sec)
```

### 文字コード

- ・ <https://yoku0825.blogspot.com/2017/04/mysql-801utf8mb4ja0900asc.html>

### 確認

```
mysql> show create database test_db;
+-----+-----+
| Database | Create Database |
+-----+-----+
| test_db  | CREATE DATABASE `test_db` /*!40100 DEFAULT CHARACTER SET utf8 */ |
+-----+-----+
1 row in set (0.00 sec)
```

## 権限

- ・ 接続を許可するユーザーを mysql データベース内で管理している
- ・ ユーザーがアクセスする際にはアクセス元のホスト (IP アドレス) もセットで認証が行われる

### 権限テーブル

テーブル名	説明
user	ユーザーの基本的な定義

host	ホストに対する権限の定義
db	データベースに対する権限の定義
tables_priv	テーブルに対する権限の定義
columns_priv	カラムに対する権限の定義

## ユーザーの作成

```
mysql> create user test_user@localhost;
Query OK, 0 rows affected (0.03 sec)
```

## 権限の付与

### 限定して付与

```
mysql> grant create,alter,select,insert,update,delete,index on *.* to test_user@localhost identified
by 'newpassword';
Query OK, 0 rows affected (0.00 sec)
```

### 管理権限を付与

- ・ test\_admin@localhost を作成した上で管理権限を付与
- ・ リモートアクセスを可能とするには、test\_admin@localhost の部分を test\_admin@'%' とする

```
mysql> grant all privileges on *.* to test_admin@localhost identified by 'newpassword' with grant
option;
Query OK, 0 rows affected (0.00 sec)
```

### 権限の確認

```
mysql> show grants for {user}
```

## 外部から接続

### ポートを開ける

```
# firewall-cmd --permanent --zone=public --add-port=3306/tcp
# firewall-cmd --reload
```

### root@% を追加

### 確認

```
mysql> select user,host from mysql.user;
+-----+-----+
| user          | host          |
+-----+-----+
| mysql.infoschema | localhost    |
| mysql.session   | localhost    |
| mysql.sys       | localhost    |
| pma             | localhost    |
| root           | localhost    |
+-----+-----+
```

wordpress	localhost

## ユーザーの追加

```
mysql> create user 'root'@'%' identified by 'パスワード';
mysql> set password for 'root'@'%' = 'パスワード';
mysql> grant all on *.* to 'root'@'%';
```

## 確認

```
mysql> select user, host from user;
```

user	host
root	%
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
pma	localhost
root	localhost
wordpress	localhost

## バックアップとリストア

### mysqldump

- ・通常 mysqldump コマンドを使用する
- ・データをテキストファイルとしてダンプするというシンプルな仕組みのコマンド
- ・CREATE TABLE および INSERT 文として出力される

### テーブル名を指定してダンプ

```
mysqldump [ オプション ] データベース名 [ テーブル名 ... ]
```

### 複数のデータベースを対象とする

```
mysqldump [ オプション ] --databases [ オプション ] データベース名 [ データベース名 ... ]
```

### すべてのデータベースを対象とする

```
mysqldump [ オプション ] --all-databases [ オプション ]
```

## 例

### バックアップ例

```
# mysqldump --default-character-set=utf8 -uroot -p mt > mt_backup20140120.sql
Enter password:
```

### リストア例

- ・mysqldump の出力は SQL 文の羅列であるため、復元するにはリダイレクトを使って mysql に結果を与えれば良い

```
# mysql -u root -p mt < mt_backup20140120.sql
```

## Tips

### 管理

#### データファイルの場所

```
/etc/my.cnf  
datadir=/var/lib/mysql
```

#### テーブル一覧を表示

```
# mysql -u root -p  
mysql> use mt  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A
```

```
Database changed  
mysql> show tables;  
+-----+  
| Tables_in_mt |  
+-----+  
| mt_asset      |  
| mt_asset_meta |  
| mt_association |  
|               |  
| mt_ts_funcmap |  
| mt_ts_job     |  
+-----+  
44 rows in set (0.00 sec)
```

#### セッションの確認と切断

```
show processlist;  
kill [Id]
```

### SQL

#### 制約を無視してテーブルを truncate する

- ・一旦、FOREIGN\_KEY\_CHECKS を 0 にする

```
mysql> SET FOREIGN_KEY_CHECKS=0;  
mysql> truncate table hoge;  
mysql> SET FOREIGN_KEY_CHECKS=1;
```

### mysql コマンド

#### ソースファイルの SQL を実行

```
¥.<ファイル名>
```

#### データベースの変更

```
use
```

## ヘルプ

help

## テーブル定義を確認

show create table テーブル名

## テーブル一覧

show tables

## ポートを調べる

```
mysql> show variables like 'port';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| port          | 3306  |
+-----+-----+
1 row in set (0.00 sec)
```

## プログラミング

### JDBC Download

- <http://dev.mysql.com/downloads/connector/>

### C# から接続

```
using MySql.Data.MySqlClient;
using System;
using System.Data;

namespace Mt2Wp
{
    class Program
    {
        static void Main(string[] args)
        {
            var pgm = new Program();
            pgm.AccessTest();
        }

        public void AccessTest()
        {
            // Unable to convert MySQL date/time value to System.DateTime
            // https://stackoverflow.com/questions/2934844
            /unable-to-convert-mysql-date-time-value-to-system-datetime
            var conInfo = $"Database=wordpress;Data Source=192.168.0.2 ;User
            Id=root;Password={password}; pooling = false; convert zero datetime=True";

            using(var conn = new MySqlConnection(conInfo))
            {
                conn.Open();

                var da = new MySqlDataAdapter("select * from wp_posts", conn);
                var ds = new DataSet();
                da.Fill(ds);

                foreach (DataTable table in ds.Tables)
                {
                    foreach (DataRow row in table.Rows)
                    {
                        foreach (DataColumn col in table.Columns)
                        {

```



```
        Console.WriteLine($"{row[col]}¥t");
    }
    Console.WriteLine("");
}
}
}
}
}
```