

# データベースシステム

## 第10回

---

理工学部情報科学科

松澤 智史

# 本日の内容

- ユーザ作成
- 権限付与・剥奪・ROLE
- 外部プログラムからの利用
- カーソル

# 新規ユーザの作成

パスワード無しの場合

```
mysql> CREATE USER user名@hostname
```

パスワード有りの場合

```
mysql> CREATE USER user名@hostname IDENTIFIED BY  
  'password'
```

Hostnameは許可する接続元を指定

※通常はlocalhostで良い

# ユーザの削除

- `mysql> DROP USER user名@hostname`

hostnameが異なる同名ユーザは削除されない

# パスワード

- MySQLのパスワード認証方式
  - caching\_sha2\_password
  - sha256\_password
  - mysql\_native\_password
- MySQL8からcaching\_sha2\_passwordがデフォルト
- PHP等の外部プログラムが対応していないと外部プログラムからは認証不可
  - ※現在PHPではcaching\_sha2\_passwordに未対応

# mysql\_native\_passwordでのユーザ作成

```
mysql> CREATE USER user名@hostname IDENTIFIED WITH  
mysql_native_password BY 'password';
```

## 確認

```
mysql> SELECT user, plugin from mysql.user;
```

```
mysql> create user test_user@localhost identified with mysql_native_password by 'password';  
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> select user, plugin from mysql.user;
```

user	plugin
matsuzawa	caching_sha2_password
another_user	caching_sha2_password
mysql.infoschema	caching_sha2_password
mysql.session	caching_sha2_password
mysql.sys	caching_sha2_password
root	caching_sha2_password
test_user	mysql_native_password

```
7 rows in set (0.00 sec)
```

# 新規ユーザの権限

- 権限確認

mysql> SHOW GRANTS FOR *user名@hostname*

```
mysql> create user test_user@localhost identified with mysql_native_password by 'password';
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> select user, plugin from mysql.user;
```

user	plugin
matsuzawa	caching_sha2_password
another_user	caching_sha2_password
mysql.infoschema	caching_sha2_password
mysql.session	caching_sha2_password
mysql.sys	caching_sha2_password
root	caching_sha2_password
test_user	mysql_native_password

```
7 rows in set (0.00 sec)
```

```
mysql> show grants for test_user@localhost;
```

Grants for test_user@localhost
GRANT USAGE ON *.* TO `test_user`@`localhost`

```
1 row in set (0.00 sec)
```

```
mysql>
```

# 新規ユーザの権限

- 作成時はまだ何も権限なしなので必要な権限を与える

GRANT 与える権限 ON データベース名. テーブル名 TO user 名@hostname;

```
mysql> grant select on test_db.* to test_user@localhost;
Query OK, 0 rows affected (0.01 sec)
```

Grants for test_user@localhost	test_dbのすべての テーブルに対して SELECTのみ許可
GRANT USAGE ON *.* TO `test_user`@`localhost`	
GRANT SELECT ON `test_db`.* TO `test_user`@`localhost`	

```
2 rows in set (0.00 sec)
```

```
mysql> grant all on test_db.* to test_user@localhost;
Query OK, 0 rows affected (0.01 sec)
```

Grants for test_user@localhost	test_dbのすべての テーブルに対して すべての操作を許可
GRANT USAGE ON *.* TO `test_user`@`localhost`	
GRANT ALL PRIVILEGES ON `test_db`.* TO `test_user`@`localhost`	

```
2 rows in set (0.00 sec)
```

```
mysql> _
```



# 権限の剥奪

- REVOKE *剥奪する権限* FROM データベース名. テーブル名 TO *user名@hostname*;

```
mysql> show grants for test_user@localhost;
```

Grants for test_user@localhost	
GRANT USAGE ON *.* TO `test_user`@`localhost`	
GRANT ALL PRIVILEGES ON `test_db`.* TO `test_user`@`localhost`	

```
2 rows in set (0.00 sec)
```

```
mysql> revoke update on test_db.* from test_user@localhost;  
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> show grants for test_user@localhost;
```

Grants for test_user@localhost	
GRANT USAGE ON *.* TO `test_user`@`localhost`	
GRANT SELECT, INSERT, DELETE, CREATE, DROP, REFERENCES, INDEX, ALTER, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, EVENT, TRIGGER ON `test_db`.* TO `test_user`@`localhost`	

```
2 rows in set (0.00 sec)
```

```
mysql>
```

# ROLE

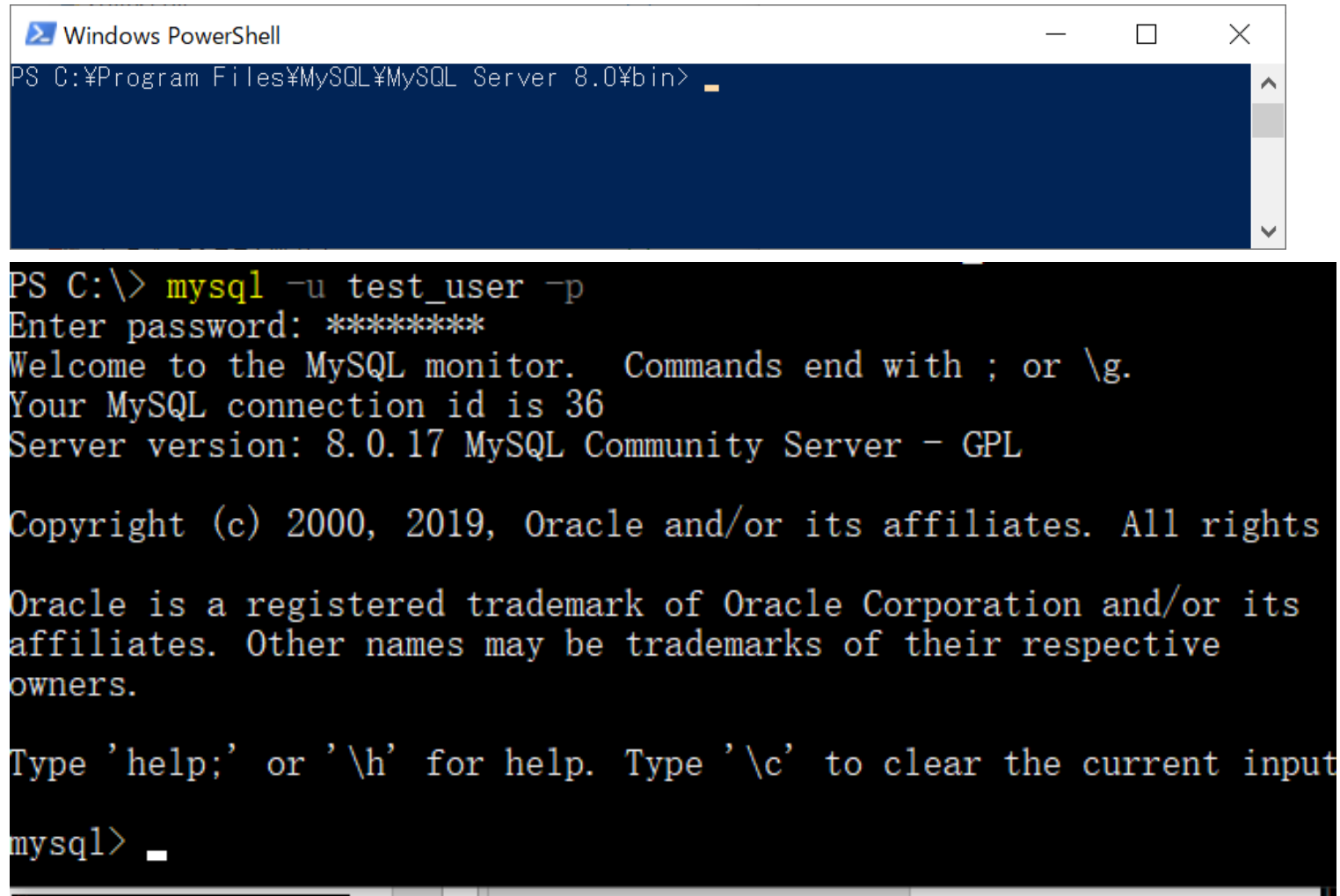
- あるユーザに細かい権限付与を行った後に別のユーザにも同じ権限を付与する場合、何度も同じ権限付与を実行することになる
- ROLEに権限付与を行っておき、ユーザにはROLEを付与することで簡略化できる

CREATE ROLE ロール名;

GRANT 権限範囲 ON テーブル TO ロール名

GRANT ロール名 TO ユーザ1, ユーザ2...

# 新規ユーザのログイン確認



```
Windows PowerShell
PS C:\Program Files\MySQL\MySQL Server 8.0\bin>

PS C:\> mysql -u test_user -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 36
Server version: 8.0.17 MySQL Community Server - GPL

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
mysql> _
```

# PHPからデータベースにアクセス

- 使用コード

```
<?php

$link = mysqli_connect('127.0.0.1', 'test_user', 'password');

if (mysqli_connect_errno()) {
    die("could not connected." . mysqli_connect_error() . "\n");
}

echo "connected successfully.\n";

$query = "SELECT id, name FROM test_db.student;";

if ($result = mysqli_query($link, $query)) {
    echo "SELECT\n";
    foreach ($result as $row) {
        var_dump($row);
    }
}

mysqli_close($link);
```

# PHPからデータベースにアクセス

- 実行結果

```
t-matsu@VersaPro ~ $ php select.php
connected successfully.
SELECT
array(2) {
    ["id"]=>
        string(1) "1"
    ["name"]=>
        string(5) "Alice"
}
array(2) {
    ["id"]=>
        string(1) "2"
    ["name"]=>
        string(3) "Bob"
}
array(2) {
    ["id"]=>
        string(1) "3"
    ["name"]=>
        string(7) "Charlie"
}
array(2) {
    ["id"]=>
```

# カーソル

- 一行ごとにデータを処理したい場合に使用する
- クエリの結果集合を一時的に蓄えておくための仮想的な作業領域をカーソルという
- 現在位置を示すポインタがある
- ループ処理が実行されるたびに一行ずつポインタが進む

# カーソルの利用手順

## 1. カーソルの宣言

```
DECLARE カーソル名 [ INSENSITIVE ] [ SCROLL ] CURSOR    FOR  
{ SELECT文 [ 更新可能性句 ] } | { 準備された文 } ;
```

## 2. カーソルを開く

```
OPEN カーソル名
```

## 3. 一行ごとにデータを取り出す(ループ処理)

```
FETCH [[NEXT] FROM] cursor_name INTO var_name [, var_name] ...
```

## 4. カーソルを閉じる

```
CLOSE カーソル名
```

# FETCHの例

FETCH student INTO A, B, C

制御ポインタのある行のデータが順にA,B,Cに格納される

studentテーブル

制御ポインタ →	1	Alice	63
	2	Bob	64
	3	Charlie	63

A= 1

B= Alice

C= 63

が格納(代入)される

※制御ポインタが1つ進むと2行目のデータがそれぞれ格納される



# カーソル(プロシージャ利用時のサンプル)

```
DELIMITER //
CREATE PROCEDURE csrtest( IN in_id INT, OUT out_str VARCHAR(10) )
BEGIN
    DECLARE myId INT;
    DECLARE myStr VARCHAR( 10 );
    DECLARE myCur CURSOR FOR SELECT id, name FROM student;
    SET @pos = 0;
    SELECT COUNT(*) INTO @total FROM student;
    SET out_str = '';
    OPEN myCur;
    WHILE @total > @pos DO
        FETCH myCur INTO myId, myStr;
        IF myId = in_id THEN
            SET out_str = myStr;
            SET @pos = @total;
        END IF;

        SET @pos = @pos +1;
    END WHILE;
    CLOSE myCur;
END
//
DELIMITER ;
```

# 実行例

- cursor\_test.sqlに前スライドのprocedureが記載されている

```
mysql> source cursor_test.sql;
Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> call csrtest(3, @str);
Query OK, 0 rows affected (0.00 sec)

mysql> select @str;
+-----+
| @str  |
+-----+
| Charlie |
+-----+
1 row in set (0.00 sec)

mysql> _
```

# 特殊な変数

SQLSTATE変数(文字列型)

主な例

‘00000’ 正常終了

‘02000’ データの終端

‘22000’ データ例外発生

‘53000’ リソース不足

# SQLSTATEを使った例

```
DELIMITER //
CREATE PROCEDURE csrtest( IN in_id INT, OUT out_str VARCHAR(10) )
BEGIN
    DECLARE done INT DEFAULT 1;
    DECLARE myId INT;
    DECLARE myStr VARCHAR( 10 );
    DECLARE myCur CURSOR FOR SELECT id, name FROM student;
    DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET done = 0;
    SET @pos = 0;
    SELECT COUNT(*) INTO @total FROM student;
    SET out_str = '';

    OPEN myCur;
    WHILE done DO
        FETCH myCur INTO myId, myStr;
        IF myId = in_id THEN
            SET out_str = myStr;
            SET @pos = @total;
        END IF;
        SET @pos = @pos + 1;
    END WHILE;

    CLOSE myCur;
END
//
DELIMITER ;
```

SQLSTATEが'02000'(終端)になると  
変数doneが0になる

# カーソルについて

- SQL文内に手続き型言語のような記述が可能
- 多くの場合呼び出し元プログラムで行ごとの処理をされるので、カーソルは使われないことも多い

```
<?php
$link = mysqli_connect('127.0.0.1', 'test_user', 'password');

if (mysqli_connect_errno()) {
    die("could not connected." . mysqli_connect_error() . "\n");
}
$sql = 'SELECT * FROM test_db.student';

$result = mysqli_query($link, $sql);
while ($data = mysqli_fetch_array($result)) {
    echo $data['id'] . ':' . $data['name'] . '<' . $data['email'] . ">\n";
}
$db = mysqli_close($link);
~
```

# Javaでの利用

- JDBCのダウンロード

- MySQL Yum Repository
- MySQL APT Repository
- MySQL SUSE Repository
- MySQL Community Server
- MySQL Cluster
- MySQL Router
- MySQL Shell
- MySQL Workbench
- MySQL Installer for Windows
- MySQL for Excel
- MySQL for Visual Studio
- MySQL Notifier
- Connector/C (libmysqlclient)
- Connector/C++
- Connector/J
- Connector/NET
- Connector/Node.js
- Connector/ODBC
- Connector/Python
- MySQL Native Driver for PHP
- MySQL Benchmark Tool
- Time zone description tables
- Download Archives

# Javaでの利用


**General Availability (GA) Releases**

**Connector/J 8.0.18**

Select Operating System:  
**Platform Independent**

Looking for previous GA versions?

<b>Platform Independent (Architecture Independent), Compressed TAR Archive</b> (mysql-connector-java-8.0.18.tar.gz)	8.0.18	3.7M	<b>Download</b>
MD5: ee3c3d8c0ca7b392c3ccc93f6eb3c165   <a href="#">Signature</a>			
<b>Platform Independent (Architecture Independent), ZIP Archive</b> (mysql-connector-java-8.0.18.zip)	8.0.18	4.4M	<b>Download</b>
MD5: 45c0dd4897ec9ba89ef3544064f1537b   <a href="#">Signature</a>			

 We suggest that you use the [MD5 checksums and GnuPG signatures](#) to verify the integrity of the packages you download.



We suggest that you use the [MD5 checksums and GnuPG signatures](#) to verify the integrity of the packages you download.

# Javaでの利用(Windows)

解凍すると

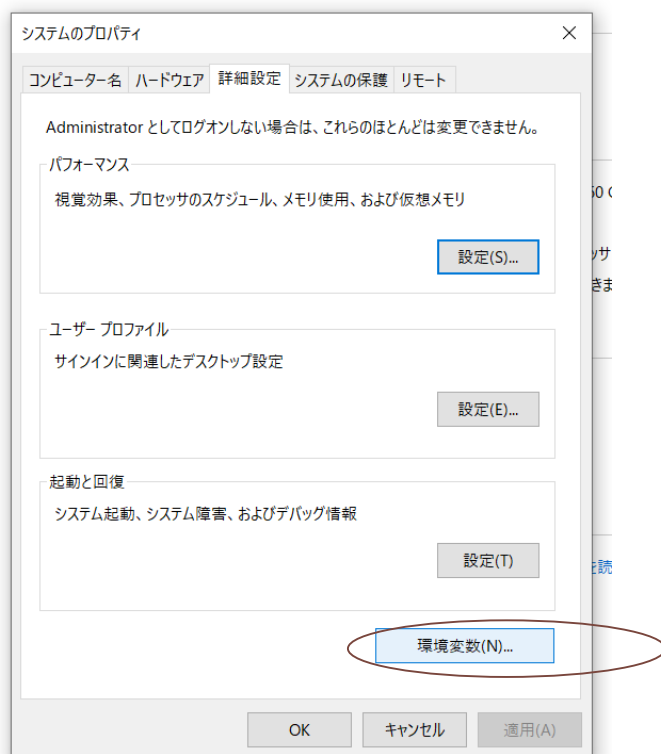
mysql-connector-java-8.0.\*\*.jar

が現れるので, このファイルにCLASSPATHを通す

コントロールパネル ホーム

コンピューターの基本的な情報の表示

- デバイス マネージャー
- リモートの設定
- システムの保護
- システムの詳細設定



```
t-matsu@VersaPro ~ $ javac JDBCTest.java
t-matsu@VersaPro ~ $ java JDBCTest
1:Alice
2:Bob
3:Charlie
4:Dave
t-matsu@VersaPro ~ $
```

関連項目

mysql-connector-java-8.0.\*\*.jar



# Javaでの利用(Linux)

```
t-matsu@ubuntu20:~$ sudo apt install ./mysql-connector-java_8.0.22-1ubuntu20.04_all.deb
パッケージリストを読み込んでいます... 完了
依存関係ツリーを作成しています
状態情報を読み取っています... 完了
注意、 './mysql-connector-java_8.0.22-1ubuntu20.04_all.deb' の代わりに 'mysql-connector-java'
を選択します
以下のパッケージが自動でインストールされましたが、もう必要とされていません:
  libfprint-2-tod1 libllvm9
これを削除するには 'sudo apt autoremove' を利用してください。
以下のパッケージが新たにインストールされます:
  mysql-connector-java
アップグレード: 0 個、新規インストール: 1 個、削除: 0 個、保留: 1 個。
2,364 kB 中 0 B のアーカイブを取得する必要があります。
```

```
t-matsu@ubuntu20:~$ ls /usr/share/java
```

commons-logging-1.2.jar	gettext.jar	libintl.jar
commons-logging-adapters-1.2.jar	java-atk-wrapper.jar	mysql-connector-java-8.0.22.jar
commons-logging-adapters.jar	java_defaults.mk	pdfbox-1.8.16.jar
commons-logging-api-1.2.jar	java_uno.jar	pdfbox.jar
commons-logging-api.jar	juh-6.4.6.jar	ridl-6.4.6.jar
commons-logging.jar	juh.jar	ridl.jar
fontbox-1.8.16.jar	jurt-6.4.6.jar	unoloader-6.4.6.jar
fontbox.iar	iurt.iar	unoloader.iar

```
t-matsu@ubuntu20:~$ javac JDBCTest.java
```

```
t-matsu@ubuntu20:~$ java -classpath /usr/share/java/mysql-connector-java-8.0.22.jar:. JDBCTest
```

.(カレントディレクトリ)を忘れないこと

# まとめ

- MySQLのユーザ管理
  - ユーザ作成・削除
  - 権限付与・剥奪, ROLE
- 外部プログラミング言語からのアクセス
- カーソル

質問あればどうぞ