

データベース操作

MySQL編

データベースの作成→テーブルの作成→データの操作まで基本的な命令文を実行して、動作を確認してみましょう！

Step 1 MySQLの起動（Windows）

コマンドプロンプトを起動させ、DBMS（MySQL）を起動させましょう！

```
mysql -u [ユーザー名] -p  
Enter password: ****
```

→コマンドプロンプト

```
C:\> mysql -u root -p
```

```
Enter password:****
```

Step 1 MySQLの起動 (Windows)

→MySQLが起動した場合

```
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 14  
Server version: 8.0.16 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 statement.  
  
mysql>
```

→MySQLが起動できなかった場合 (ユーザー名またはパスワードが違う)

```
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
```

→MySQLが起動できなかった場合 (ユーザー名またはパスワードが違う)

```
内部コマンドまたは外部コマンド、  
操作可能なプログラムまたはバッチ ファイルとして認識されていません。
```

SQLの命令文の種類について

/* データベースの3種類の命令文 */

DDL (Date Definition Language)・・・データの定義言語

データを格納する場所を作成 (CREAT)、変更 (ALTER)、削除 (DROP) などを行う命令文

DML (Date Manipulation Language)・・・データの操作言語

格納されたデータを挿入 (INSERT)、抽出 (SELECT)、更新 (UPDATE)、
削除 (DELETE) など行う命令文

DCL (Date Control Language)・・・データの制御言語

DDL、DMLの利用に関する権限の許可 (GRANT) や剥奪 (REVOKE) などを設定する命令文

Step2 データベースの作成（CREATE文）

-- 使うデータベースを作成する

CREATE DATABASE [データベース名];

--以下のデータベースを作成する

	論理名	物理名
データベース名	サンプルデータベース	sample_db

→SQL

```
mysql> CREATE DATABASE sample_db;
```

→実行結果 データベースの作成に成功した場合

```
mysql> create database sample_db;  
Query OK, 1 row affected (0.06 sec)
```

※「SHOW DATABASES」で、データベースが作成できているか確認してください。

エラーについて

[エラーポイント1]

```
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'database' at line 1
```

※俗に、syntaxエラーと呼ばれるエラーです

- SQLのコードに問題がある、記述に間違いがある場合
- 名前に「予約語」使われた場合

↓対応

「syntax to use near」の後のシングルクォーテーション付近がエラー箇所ですので修正してください。

[エラーポイント2]

```
ERROR 1007 (HY000): Can't create database 'sample_db'; database exists
```

※同じ名前を付けたデータベースを作成しようとした場合のエラーです。同じ名前のデータベースは作成できません。

↓対応

データベース名を変更して作成してください。

注意ポイント

※入力待ち状態が続く場合

```
>
```

↓対応
続けて入力するか、「;」セミコロンを入力して終了してください。

```
>;
```

※エラーが出る場合もあります。もう一度SQLを入力しましょう！

※入力待ち状態が続く場合

```
'>
```

↓対応
「'」シングルクォーテーションを入力してEnterしてください。

```
>
```

※入力待ち状態に変わります。

Step3 データベースの選択（USE文）

-- 使うデータベースを選択します。

USE [データベース名];

→SQL

```
mysql> USE sample_db;
```

→実行結果

```
Database changed
```

Step4 テーブルの作成（CREATE文）

-- データを管理するためのテーブルを作成する

CREATE TABLE [テーブル名] (列名 データ型, 列名 データ型, ...);

-- 以下のテーブルを作成する

	論理名	物理名
テーブル名	商品	product

論理名	物理名	データ型	Null	デフォルト値	制約
商品ID	id	char(5)	no		primary key
商品名	name	varchar(100)	no		
高さ	height	Int	no	0	0以上
幅	width	Int	no	0	0以上
重さ	weight	Int	no	0	0以上
価格	price	Int	no		0以上

データ型と制約について

--- データ型 ---

テーブルには格納するデータを限定するために「データ型」を指定する必要がある

主なデータ型

- char型 … 固定長の文字列データを格納することができる
- varchar型…可変長の文字列データを格納することができる
- int型…整数値データを格納することができる

などがあります。

--- 制約 ---

格納するデータに、一定の「制約」を設けることでより安全にデータを管理することができる

主な制約

- NOT NULL…NULLの格納を禁止する
- PRIMARY KEY…重複するデータの格納を禁止する
- CHECK…格納できるデータに条件を付ける
- DEFAULT…デフォルト値を設定する

などがあります。

Step4 テーブルの作成（CREATE文）

→SQL

```
mysql> CREATE TABLE product (  
id          char(5)      NOT NULL    PRIMARY KEY ,  
name        varchar(100) NOT NULL ,  
height      int          NOT NULL    DEFAULT 0 ,  
width       int          NOT NULL    DEFAULT 0 ,  
weight      int          NOT NULL    DEFAULT 0 ,  
price       int          NOT NULL    CHECK(price >= 0)  
);
```

→実行結果 ※テーブルの作成に成功した場合

```
Query OK, 1 row affected (0.04 sec)
```

Step5 テーブルの構造確認（SHOW文）

-- テーブルの構造を確認する

SHOW COLUMNS FROM [テーブル名]; または DESC [テーブル名];

→SQL

```
SHOW COLUMNS FROM product;
```

または、

```
DESC product;
```

Step6 テーブルにデータを挿入する（INSERT文）

INSERT INTO [テーブル名] (列名1, 列名2, 列名3, …)
VALUES(データ1, データ2, データ3, …);

以下のデータを挿入する

id	name	height	width	weight	price
s0001	机A	800	800	10	12000

→SQL

```
INSERT INTO product( id, name, height, width, weight, price )  
VALUES('s0001', '机A', 800, 800, 10, 12000 );
```

→実行結果 ※データの挿入に成功した場合

```
Query OK, 1 row affected
```

Step7 テーブルのデータを確認する（SELECT文）

SELECT * FROM [テーブル名];

→SQL

```
SELECT * FROM product;
```

→結果

```
+-----+-----+-----+-----+-----+-----+  
| id   | name | height | width | weight | price |  
+-----+-----+-----+-----+-----+-----+  
|s0001| 机A  |  800  |  800  |   10   |12000 |  
+-----+-----+-----+-----+-----+-----+
```

-- 練習 --

以下のデータをINSERT文で、producテーブルに格納し、SELECT文で確認してください。

id	name	height	width	Weight	price
s0002	机B	800	1000	15	15000

Setp8 複数行のデータを挿入する（INSERT文）

INSERT INTO [テーブル名] (列名1, 列名2, 列名3, …)
VALUES(データ1, データ2, データ3, …), (データ1, データ2, データ3, …) ;

以下のデータを挿入する

id	name	height	width	Weight	price
s0003	椅子A	600	450	5	6000
s0004	椅子B	600	450	5	0

→SQL

```
INSERT INTO product(id, name, height, width, weight, price)
VALUES
('s0003', '椅子A', 600, 450, 5, 6000) ,
('s0004', '椅子B', 600, 450, 5, 0) ;
```

→結果

```
Query OK, 2 row affected
```

※データが挿入されているかSELECT文で確認してください。

Step9 列を指定してデータを確認する（SELECT文）

SELECT [列A] , [列B] , [列C] FROM [テーブル名] ;

→SQL

```
SELECT id , name , price FROM product ;
```

→結果

id	name	price
s0001	机A	12000
s0002	机B	15000
s0003	椅子A	6000
s0004	椅子B	0

Step10 行を絞り込んでデータを抽出する（SELECT文）

SELECT * FROM [テーブル名] WHERE [絞り込み条件] ;

※条件の絞り込みにはWHERE句を使います。

Idが「s0004」を絞り込み条件としてデータを抽出する

→SQL

```
SELECT * FROM product WHERE id = 's0004' ;
```

→結果

id	name	height	width	weight	price
s0004	椅子B	600	450	5	0

Step11 データの更新（UPDATE文）

テーブルのデータを更新する

UPDATE [テーブル名] SET [更新したい列] = [更新データ] WHERE [絞り込み条件];

以下のデータを元に、Id列が「s0004」の行のpriceを「7000」に更新する

id	name	height	width	weight	price
s0004	椅子B	600	450	5	7000

→SQL

```
UPDATE product SET price = 7000 WHERE id = 's0004';
```

→実行結果 ※データの更新に成功した場合

```
Query OK, 1 row affected
```

※データが更新されたかSELECT文で確認してください。

Step12 行を絞り込んでデータを削除する（DELETE文）

DELET FROM [テーブル名] WHERE [絞り込み条件];

Idが「s0004」を条件として、データを削除する

→SQL

```
DELETE FROM product WHERE id = 's0004';
```

→実行結果 ※データの削除に成功した場合

```
Query OK, 1 row affected
```

※データが削除されているかSELECT文で確認してください。

Step13 テーブル内の全てのデータを削除する

DELETE FROM [テーブル名];

productテーブルのデータを全て削除する

→SQL

```
DELETE FROM product ;
```

→実行結果 ※データの削除に成功した場合

```
Query OK, 3 row affected
```

※データが削除されているかSELECT文で確認してください。

Step14 テーブルを削除する（DROP文）

DROP TABLE [テーブル名];

productテーブルを削除する

→SQL

```
DROP TABLE product ;
```

→実行結果 ※テーブルの削除に成功した場合

```
Query OK, 0 row affected
```

※「SHOW TABLES」で、テーブルが削除されているかを確認してください。

Step15 データベースを削除する（DROP文）

DROP DATABASE [データベース名];

sample_dbデータベースを削除する

→SQL

```
DROP DATABASE sample_db;
```

→実行結果 ※データベースの削除に成功した場合

```
Query OK, 0 row affected
```

※「SHOW DATABASES」で、データベースが削除されているかを確認してください。

Step16 MySQLを終了する

→SQL

```
mysql> exit
```

または、

```
mysql> quit
```

または、ctrl + z を入力してEnter