

はじめての MySQL(XAMPP 版) ver1.5

Seiichi Nukayama

2025 年 9 月 29 日

目次

1	MySQL にログインする	1
1.1	root(管理者) でログインする	1
2	データベースを設計する	1
2.1	扱うデータ	1
2.2	どのような表をつくるか?	2
2.3	primary key	2
3	データベースを作成する	3
3.1	データベースの作成	3
3.2	データベースの確認	3
3.3	データベースの使用宣言	4
4	テーブルを定義する	4
4.1	テーブルの定義	4
4.2	テーブルの確認	5
5	データの挿入	5
6	データの表示	6
6.1	一覧表示	6
6.2	抽出して表示	7
7	データの修正	7
8	データの削除	8
9	CRUD	8
10	練習問題	9

1 MySQL にログインする

1.1 root(管理者) でログインする

データベースを利用するためには、まず、そのデータベースを管理している人(管理者)から、アカウント(ユーザー名とパスワード)を発行してもらわなくてはならない。

そして、通常は1つのデータベースが与えられ、そのデータベースの中に複数のテーブル(表)を作成していくことになる。

しかし、ここでは管理者のままでMySQLデータベースを操作していくことにする。そして、操作に慣れたら、一般ユーザーを作成し、一般ユーザーとしてデータベースを操作していくことにする。

管理者(root)でのログイン

```
> mysql -u root -p
Enter password: (何も入力せずに Enter)
```

xamppをインストールした場合、MySQL(MariaDB)にばパスワードは設定されていない。もし、パスワードを設定すれば、ここでパスワードを入力することになる。

パスワードが設定されている場合、以下のようにパスワードも含めて、一行で入力することもできる。パスワードは-pのあと、空白をはさまずに続けて書く。(パスワードが"s3cret"の場合はこうなる)

```
mysql> mysql -u root -ps3cret*1
```

2 データベースを設計する

2.1 扱うデータ

以下のようなデータを扱うこととする。

菅原文太 40 歳 1933 年生まれ 総務部	千葉真一 34 歳 1939 年生まれ 営業部	北大路欣也 30 歳 1943 年生まれ 経理部	梶芽衣子 26 歳 1947 年生まれ 営業部
----------------------------------	----------------------------------	-----------------------------------	----------------------------------

あなたがプログラマで、上のような社員名簿アプリを作成することになったとする。PHP か Java でアプリを作成することになる。クライアントの会社の総務部がこのアプリを使うことになる。そのアプリには社員の登録画面、一覧画面、編集画面、削除画面などがあるだろう。そういった画面と処理をあなたは作らなければならない。

そのときに、データを保存するしくみとして、データベースを使うことになる。かりに PHP でプログラミングするならば、PHP という言語を使ってデータベースを操作することになる。

^{*1} mysql コマンドは、"C:¥xampp¥mysql¥bin" の中の "mysql.exe" のことである。このフォルダには、他にも "mysqldump.exe" などいろいろなコマンドが置かれている。

2.2 どのような表をつくるか?

データベースは表の形でイメージすることができる。しかし、上記のデータを見て、それをそのまま表にしてはいけない。

	A	B	C	D	E
1	菅原文太	千葉真一	北大路欣也	梶芽衣子	
2	40	34	30	26	
3	1933	1939	1943	1947	
4	総務部	営業部	経理部	営業部	
5					

この表は、1 件のデータが縦に配置されて、それが人数分横に続いている。これは良くない。
次の表のように、1 件のデータを横に配置する。

	A	B	C	D	E
1	名前	年齢	誕生年	部署	
2	菅原文太	40	1933	総務部	
3	千葉真一	34	1939	営業部	
4	北大路欣也	30	1943	経理部	
5	梶芽衣子	26	1947	営業部	
6					
7					

そして、縦には同じ種類のデータが並ぶ。だから、それぞれの列には、その列の内容を表す項目名をつけることができる。

この列のことを **カラム (項目)** という。(フィールドともいう)

そして、1 件のデータを表す横 1 行を **レコード** という。この表には 4 件のデータがあり、カラムは 4 である。

しかし、これだけではデータベースにはならない。各レコードには、そのレコードの独自性を保証するデータが必要なのである。それを **プライマリー・キー** という。

2.3 primary key

データベースにデータを格納する際には、そのデータに **primary key (独自キー)** が必要となる。primary key とは、そのデータを他と区別するためのデータである。菅原文太というデータは、この 4 つの中では独自であるが、他のデータを追加する際に、同じデータに出会う可能性 (同姓同名) を排除できない。さらに日本語である以上、文字コードの問題を避けることもできない。つまり、同じ菅原文太という文字でも UTF-8 と Shift_JIS では別物と判定されるのである。

となると、この 4 つのデータには **primary key** となるものがないということになる。

このような場合、データベースの設計者が **primary key** を追加することになる。ここでは 数字を primary key として追加する。つまり、菅原文太は 1、千葉真一は 2 というふうにする。

そして、その項目名をここでは **id** とした。

	A	B	C	D	E	F
1	id	名前	年齢	誕生年	部署	
2	1	菅原文太	40	1933	総務部	
3	2	千葉真一	34	1939	営業部	
4	3	北大路欣也	30	1943	経理部	
5	4	梶芽衣子	26	1947	営業部	
6						
7						

primary key には、数字やコードが使われる。脚注^{*2}

3 データベースを作成する

3.1 データベースの作成

"rensyu" というデータベースを作成する。

```
MariaDB [(none)]> create database rensyu <Enter キー>
```

このように入力すると、以下のようになる。

```
MariaDB [(none)]> create database rensyu
->
```

これは、入力の終わりがまだないので、次の入力を受け付けているのである。

入力の終わりは ";"(セミコロン) あるいは "\g" である。

```
MariaDB [(none)]> create database rensyu
-> ; <Enter キー>
```

";"(セミコロン) あるいは "\g" を入力して <Enter キー> を押す。また、 "\c" を入力して" キャンセル" することもできる。

3.2 データベースの確認

データベースがちゃんと作成できたか、確認する。

```
MariaDB [(none)]> show databases; (複数形)
```

```
+-----+
| Database |
+-----+
```

^{*2} 整数や固定長の文字列がよく使われる。ただし、社員コードや ISBN などは、データベース外部の都合で変更されたりすると、データベースも変更せざるを得なくなるので、プライマリキーとしてはよろしくないと言える。

information_schema
rensyu
+-----+

3.3 データベースの使用宣言

まず、使用宣言を行う。

```
MariaDB [(none)]> use rensyu;
```

Database changed と表示される。

4 テーブルを定義する

4.1 テーブルの定義

以下のようなテーブルを作成することとする。

表 1 emp

ID	名前	年齢	誕生年	部署
1	菅原文太	40	1933	総務部
2	千葉真一	34	1939	営業部
3	北大路欣也	30	1943	経理部
4	梶芽衣子	26	1947	営業部

それぞれの列のデータ型を決める。

列	データの種類	データ型
ID	整数	int 型
名前	文字列 (可変長)	varchar 型
年齢	整数	int 型
誕生年	年	year 型
部署	文字列 (可変長)	varchar 型

この表は以下のように定義できる。表の名前を "emp" とする。

リスト 1 emp テーブルの定義

```
1 MariaDB [rensyu]> create table emp (
2   -> id int primary key,
3   -> name varchar(20),
4   -> age int,
5   -> birthyear year,
6   -> dept varchar(20)
```

```

7      -> );
8 Query OK, 0 row affected (0.015sec)

```

4.2 テーブルの確認

テーブルができたかどうかは、以下のコマンドで確認できる。

```
MariaDB [rensyu]> show tables; (複数形)
```

```

+-----+
| Tables_in_rensyu |
+-----+
| emp               |
+-----+

```

また、そのテーブルの定義の確認は、以下のコマンドでできる。

```
MariaDB [rensyu]> desc emp;
```

```

+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)       | NO   | PRI | NULL    |       |
| name       | varchar(20)   | YES  |     | NULL    |       |
| age        | int(11)       | YES  |     | NULL    |       |
| birthyear  | year(4)       | YES  |     | NULL    |       |
| dept       | varchar(20)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+

```

5 データの挿入

それでは、1 件分のデータを入力する。

入力データ

id	name	age	birthyear	dept
1	菅原文太	40	1933	総務部

```

1 MariaDB [rensyu]> insert into emp
2   -> (id, name, age, birthyear, dept)
3   -> values
4   -> (1, '菅原文太', 40, 1933, '総務部');
5 Query OK, 1 row affected (0.001 sec)

```

続いて、2 つめのデータを入力する。

入力データ

id	name	age	birthyear	dept
2	千葉真一	34	1939	営業部

全項目を入力する場合、項目指定を省略できる。

```
1 MariaDB [rensyu]> insert into emp
2   -> values
3   -> (2, '千葉真一', 34, 1939, '営業部');
4 Query OK, 1 row affected (0.001 sec)
```

残りの2件を一度に入力する。

入力データ

id	name	age	birthyear	dept
3	北大路欣也	30	1943	経理部
4	梶芽衣子	26	1947	営業部

```
1 MariaDB [rensyu]> insert into emp
2   -> values
3   -> (3, '北大路欣也', 30, 1943, '経理部'),
4   -> (4, '梶芽衣子', 26, 1947, '営業部');
5 Query OK, 2 rows affected (0.003 sec)
6 Records: 2 Duplicates: 0 Warnings: 0
```

6 データの表示

6.1 一覧表示

今までに入力したデータの一覧を表示する。

```
1 MariaDB [rensyu]> select * from emp;
```

あるいは、次のように出力する項目を指定できる。^{*3}

```
1 MariaDB [rensyu]> select
2   -> id,
3   -> name,
4   -> age,
5   -> birthyear,
6   -> dept
7   -> from emp;
```

id	name	age	birthyear	dept
1	菅原文太	40	1933	総務部
2	千葉真一	34	1939	営業部
3	北大路欣也	30	1943	経理部
4	梶芽衣子	26	1947	営業部

4 rows in set (0.000 sec)

^{*3} ここでは全項目を指定しているが、必要な項目だけに絞ることもできる。

6.2 抽出して表示

年齢が 30 才以上の人を抽出する。

```
1 MariaDB [rensyu]> select
2   -> *
3   -> from emp
4   -> where
5   -> age >= 30;
```

id	name	age	birthday	dept
1	菅原文太	40	1933	総務部
2	千葉真一	34	1939	営業部
3	北大路欣也	30	1943	経理部

所属が" 営業部" である人を抽出する。

```
1 MariaDB [rensyu]> select
2   -> *
3   -> from emp
4   -> where
5   -> dept like '営業%';
```

id	name	age	birthyear	dept
2	千葉真一	34	1939	営業部
4	梶芽衣子	26	1947	営業部

7 データの修正

データの修正 (更新) をしてみる。ここでは、千葉真一の 部署を" 開発部" に変更してみる。

```
1 MariaDB [rensyu]> update emp
2   -> set
3   -> dept = '開発部',
4   -> where
5   -> id = 2;
```

```
MariaDB [rensyu]> select * from emp;
```

id	name	age	birthyear	dept
1	菅原文太	40	1933	総務部
2	千葉真一	34	1939	開発部
3	北大路欣也	30	1943	経理部
4	梶芽衣子	26	1947	営業部

4 rows in set (0.000 sec)

8 データの削除

データを 1 件削除する。ここでは、北大路欣也を削除してみる。

```
1 MariaDB [rensyu]> delete from emp
2   -> where
3   -> id = 3;
```

```
MariaDB [rensyu]> select * from emp;
+----+-----+-----+-----+-----+
| id | name      | age | birthyear | dept      |
+----+-----+-----+-----+-----+
| 1  | 菅原文太  | 40  | 1933      | 総務部    |
| 2  | 千葉真一  | 34  | 1939      | 開発部    |
| 4  | 梶芽衣子  | 26  | 1947      | 営業部    |
+----+-----+-----+-----+-----+
4 rows in set (0.000 sec)
```

9 CRUD

データの挿入 (作成)(insert)、表示 (読み込み)(select)、修正 (更新)(update)、削除 (delete) は基本処理である。
Create Read Update Delete という。

10 練習問題

- (1) "ronin" という名前のデータベースを作成してください。
- (2) そのデータベースに actors というテーブルを作成し、その内容として、以下のデータを格納できるように、テーブル定義をしてください。誕生日は DATE 型にしてください。プライマリキーも設定してください。
- (3) 以下のデータを cast に登録してください。

リスト2 データ

俳優名,	性,	誕生日,	出身,	所属
-----+-----+-----+-----+				
原田芳雄,	m,	1940-02-29,	東京都,	ギルドB
勝新太郎,	m,	1931-11-29,	千葉県,	勝プロ
樋口可奈子,	f,	1958-12-13,	新潟県,	ユマニテ
石橋蓮司,	m,	1941-08-09,	東京都,	劇団第七病棟
田中邦衛,	m,	1932-11-23,	岐阜県,	俳優座
杉田かおる,	f,	1964-11-27,	東京都,	オフィスPSC
-----+-----+-----+-----+				

- (4) 出身が" 東京" である人を抽出して表示してください。
- (5) 田中邦衛さんの所属を俳優座から" なし" に変更してください。
- (6) 以下のデータを追加してください。

リスト3 追加データ

中尾彬, m, 1942-08-11, 千葉県, 古館プロジェクト

- (7) 石橋蓮司さんのデータを削除してください。