

テーブル管理

データベースが完成したので、あとはテーブルを作成するだけです。まずは定番の「user」テーブル、つまり全ユーザーを格納するテーブルを作成しましょう。

テーブルは各が新しいエントリであるテーブルなので、ここでは「ユーザー」テーブルの各が新しいユーザーになります。

```
テーブル users を作成します( id int, ユーザー名  
varchar(50), メール varchar(255), パス  
ワードテキスト, アクティブ bool,  
createdAt datetime )。
```

```
-- クエリの終了を示す「;」、-- クエリ内のスキップしても問題ありま  
せん。
```

このクエリを詳しく見てみましょう。

- 「CREATE TABLE」は説明が不要です。 - 「users」はテーブルの名前です。
- 括弧内はテーブルの列です。各列には名前と型が付けられています。ここでは主な型を見ていきます。すべての型は公式ドキュメントまたはhttps://www.w3schools.com/mysql/mysql_datatypes.aspで確認できます。

列を見てみましょう:

- 「id」はほとんどのテーブルに存在し、への参照とインデックスとして機能します。ここでは、整数を表す「int」型を指定します。「username」は「varchar(50)」型で、最大50文字の文字列を表します。
- 「email」は「varchar(255)」型で、varcharの最大文字数は255です。「password」は「text」型で、制限が
- はるかに大きい文字列です（ハッシュ化されたパスワードは基本パスワードよりもはるかに長くなる可
- 能性があるため、これは便利です）。「active」は「bool」型で、ブール値です。データベースでは「1」または「0」で表されます。
- 「createdAt」は「datetime」型で、日付と時刻を受け取ります。
-
-

とはいえ、このSQLクエリには多くのことを忘れていました。そこで、このテーブルを削除してやり直しましょう。

```
DROP TABLE ユーザー;
```

次に、いくつかのパラメータを追加します。

```
CREATE TABLE users ( id int  
NOT NULL AUTO_INCREMENT,
```

```
ユーザー名varchar(50) NOT NULL、メールアドレスvarchar(255) NOT NULL、パスワードテキストNOT NULL、アクティブbool  
DEFAULT 0、createdAt datetime  
DEFAULT CURRENT_DATE()、  
主キー (ID) 、  
ユニーク(ID)  
ユニーク (メール)  
);
```

ここで何が新しく見えるでしょうか？

- 「NOT NULL」が複数回表示されている場合、このフィールドの各エントリを入力する必要があり、空にすることはできないことを示します。
- 「id」の「AUTO_INCREMENT」は、この数値フィールドにデータが提供されない場合、新しいエントリごとにこの数値フィールドが「1」ずつ増加することを示します。
- 「DEFAULT」を使用すると、何も指定されていない場合にデフォルト値を指定できます。
- 「CURRENT_DATE()」は現在の日付を取得する関数です。
- 「PRIMARY KEY(id)」は、「id」フィールドがテーブルの「PRIMARY KEY」であることを示します。つまり、このフィールドがインデックス作成に使用されます。つまり、このフィールドが各エントリを識別するフィールドです。
- 「UNIQUE()」が2回出現し、括弧内のフィールドが「UNIQUE」でなければならないことを示しています。つまり、テーブル内に既に存在する値をこのフィールドに入力しようとすると、エラーが返されます。

次のようにしてテーブルの構造を確認できます。

```
SHOW CREATE TABLE ユーザー;
```

テーブル作成を可能にするクエリが表示されます。ここで以下を実します。

```
テーブルを表示;
```

現在のデータベースに含まれるテーブルのリストが表示されます。

しかし、なんと、表に間違いがあったことが判明しました。

「CURRENT_DATE()」と入力しましたが、これは「datetime」フィールドなので、日付しか取得できず、デフォルトで「00:00:00」となる時刻は取得できません。もちろん、全てを削除して最初から作り直すこともできますが、面倒ですし、テーブルに既にデータがある場合は全て失われてしまいます。

幸いなことに、解決策があります。

```
ALTER TABLE users MODIFY COLUMN createdAt datetime DEFAULT CURRENT_TIMESTAMP;
```

- 「ALTER TABLE」は、テーブルを変更することを示し、次にどのテーブルを変更するかを指定します。
- 「MODIFY COLUMN」は、列の1つを変更することを示します。次に、変更する列を指定し、新しい属性を設定します。
- 「CURRENT_TIMESTAMP」は現在のタイムスタンプを提供し、そこから時刻と日付を抽出できます。

もう一つの変更点は、ユーザーの生年月日を最終的に知りたいという点です。

```
ALTER TABLE users ADD birthday date NOT NULL;
```

- 「ALTER TABLE」の後の「ADD」を使用すると、新しい列を追加できます。

結局のところ、年齢は私に合わないのです。

```
ALTER TABLE users DROP COLUMN birthday;
```

- 「DROP COLUMN」を使用すると、列の 1 つを削除できます。