
MySQL

コマンドプロンプトによるCRUD

2 時間目

テーブルの作成と削除方法

対応するSQL文	操作内容/意味/活用事例
Create文	データベースにテーブルを作成する。
Dropt文	作成したテーブルを削除する。

テーブルの操作方法

データベース操作は、大きく4つに分かれ
CRUD（クラッド）と呼ばれる。

名称	対応するSQL文	操作内容/意味/活用事例
Create	insert文	何らかの情報（データ）をテーブルに書き込む。 FaceBookに新規でアカウントを作る際に自分の名前やメールアドレスを登録する際（アカウント作成の際）には、FaceBookのシステムの裏側で、insert文が走っている。
Read	select文	テーブルに既に入っている何らかの情報（データ）を呼び出す。 FaceBookに既に登録しており（アカウントを持っている）、ログインした際に、自分のプロフィール写真や、自分の過去の投稿内容などが表示されるが、その裏側ではselect文が走っている。
Update	update文	テーブルに既に入っている何らかの情報（データ）を上書きする。 FaceBookに既に登録しており（アカウントを持っている）、登録済みのメールアドレスを変更したい場合などに、その裏側でupdate文が走っている。
Delete	delete文	テーブルに既に入っている何らかの情報（データ）を削除する。 FaceBookに既に登録しているアカウントを削除（退会など）する際に、その裏側でdelete文が走っている。

- 1 **create文**でデータベースを作成
- 2 **insert文**でデータの挿入

データベースの作成 (create文)

```
mysql> create database lesson02;  
Query OK, 1 row affected (0.00 sec)
```

① 『create database lesson02;』 と入力。
create文を使ってlesson02というデータベースを作成するという意味

```
mysql> show databases;
```

② 『show databases;』 と入力。
showコマンドを使ってデータベース一覧を表示するという意味。

```
+-----+  
| Database |  
+-----+  
| information_schema |  
| akira |  
| lesson01 |  
| lesson02 |  
| mysql |  
| performance_schema |  
| sys |  
| umile |  
+-----+  
8 rows in set (0.00 sec)
```

③ データベース一覧が表示された。

使用するデータベースへ移動(useコマンド)

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| akira |
| lesson01 |
| lesson02 |
| mysql |
| performance_schema |
| sys |
| umile |
+-----+
8 rows in set (0.00 sec)
```

① 『 use lesson02; 』 と入力

```
mysql> use lesson02;
```

```
Database changed
```

```
mysql>
```

② 『Database changed』 と表示された。
データベース「lesson02」に移動したという意味。

テーブルの作成 (create文)

```
mysql> create table item(  
-> item_id int,  
-> item_name varchar(255),  
-> item_team enum('spring','summer','fall','winter'),  
-> price decimal(9,2)  
-> );
```

Query OK, 0 rows affected (0.01 sec)

mysql> █

① 『create table item(
item_id int,
item_name varchar(255),
item_team enum('spring','summer','fall','winter'),
price decimal(9,2)
); 』と入力。

itemという名前のテーブルを作成するという意味。

decimal は金額のデータ型としてよく利用する。

※(9,2)は、扱う数字の桁数が全体で「9桁」、
小数点以下の桁数が「2桁」まで利用できるという意味。

enumは () 内で指定した値のみ限定する場合に使用

テーブルの構造を確認 (descコマンド)

- ① 『desc item』 と入力。
descコマンドを使ってテーブル構造を確認。

```
mysql> desc item;
```

Field	Type	Null	Key	Default	Extra
item_id	int(11)	YES		NULL	
item_name	varchar(255)	YES		NULL	
item_team	enum('spring','summer','fall','winter')	YES		NULL	
price	decimal(9,2)	YES		NULL	

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

```
mysql>
```

- ② テーブル構造が表示された。
※ これらのフィールドについては次の章で学習します。

データの挿入 (insert文)

```
mysql> insert into item(item_id, item_name, item_team, price) values  
-> (1, 'りんご', 'summer', 150),  
-> (2, 'ぶどう', 'fall', 200),  
-> (3, 'さつまいも', 'fall', 250),  
-> (4, 'ネギ', 'winter', 190),  
-> (5, 'にんじん', 'spring', 50),  
-> (6, 'トマト', 'summer', 100);  
Query OK, 6 rows affected (0.01 sec)  
Records: 6  Duplicates: 0  Warnings: 0
```

『insert into item(item_id, item_name, item_team, price) values
(1, 'りんご', 'summer', 150),
(2, 'ぶどう', 'fall', 200),
(3, 'さつまいも', 'fall', 250),
(4, 'ネギ', 'winter', 190),
(5, 'にんじん', 'spring', 50),
(6, 'トマト', 'summer', 100); 』と入力。

insert文で、テーブルで指定したid番号、商品名、季節、値段の順にデータを登録。

select文でデータの抽出(=呼び出し)

itemテーブルに入れたデータを

select文で抽出（=呼び出し）

select文でデータの抽出(=呼び出し)

- ① 『select * from item』 と入力。
* は、all (=全て) という意味。
select * from テーブル名 で全データを呼び出すという意味。

```
mysql> select * from item;
```

item_id	item_name	item_team	price
1	りんご	summer	150.00
2	ぶどう	fall	200.00
3	さつまいも	fall	250.00
4	ネギ	winter	190.00
5	にんじん	spring	50.00
6	トマト	summer	100.00

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

```
mysql>
```

- ② itemテーブルに入っていたデータが全て呼び出された。

select文でデータの抽出(=呼び出し)

```
mysql> select item_name from item;
```

① 『select item_name from item』 と入力。
item_nameの列のデータを呼び出すという意味。

item_name
りんご
ぶどう
さつまいも
ネギ
にんじん
トマト

② item_nameの列に入っていたデータが全て呼び出された。

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

select文でデータの抽出(=呼び出し)

```
[mysql> select item_name,price from item;
```

item_name	price
りんご	150.00
ぶどう	200.00
さつまいも	250.00
ネギ	190.00
にんじん	50.00
トマト	100.00

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

① 『select item_name,price from item』 と入力。
item_nameとpriceの列のデータを呼び出すという意味。

② item_nameとpriceの列に入っていたデータが全て呼び出された。

update文でデータの上書き(=更新)

itemテーブルに入れたデータを

update文で上書き（=更新）

update文でデータの上書き(=更新)

```
+-----+-----+-----+-----+
| item_id | item_name   | item_team | price  |
+-----+-----+-----+-----+
|      1 | りんご      | summer   | 150.00 |
|      2 | ぶどう      | fall     | 200.00 |
|      3 | さつまいも  | fall     | 250.00 |
|      4 | ネギ        | winter   | 190.00 |
|      5 | にんじん    | spring   |  50.00 |
|      6 | トマト      | summer   | 100.00 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

```
mysql> update item set item_team = "winter" where item_id = 1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

① 『update item set item_team="winter" where item_id=1;』 と入力。

item_idが1のデータのitem_team部分を『winter』に上書きするという意味。

② 『Query OK,』と表示されれば更新完了。

データの上書き、変更 (Update)

```
mysql> select * from item;
```

- ① 『select * from item;』 と入力。
select文で前ページのupdateの内容が更新されたかを確認。

item_id	item_name	item_team	price
1	りんご	winter	150.00
2	ぶどう	fall	200.00
3	さつまいも	fall	250.00
4	ネギ	winter	190.00
5	にんじん	spring	50.00
6	トマト	summer	100.00

6 rows in set (0.00 sec)

- ② 『winter』 に変わった。
問題なく更新されているというのがわかった。

update文でデータの上書き(=更新)

```
[mysql> update item set price = 500 where item_id = 1;  
Query OK, 1 row affected (0.00 sec)  
Rows matched: 1  Changed: 1  Warnings: 0
```

② 『Query OK,』と表示されれば更新完了。

① 『update item set price = 500 where item_id=1;』と入力。

item_idが1のデータのprice部分を『500』に上書きするという意味。

```
[mysql> select * from item;
```

item_id	item_name	item_team	price
1	りんご	winter	500.00
2	ぶどう	fall	200.00
3	さつまいも	fall	250.00
4	ネギ	winter	190.00
5	にんじん	spring	50.00
6	トマト	summer	100.00

③ 『select * from item;』と入力。
select文でupdateの内容が更新されたかを確認。

④ 『500』に変わった。

itemテーブルに入れたデータを

delete文で削除

delete文でデータの削除

```
mysql> delete from item where item_id = 1;  
Query OK, 1 row affected (0.00 sec)
```

① 『delete from item where item_id = 1;』 と入力。
item_idが1のデータを削除するという意味。

② 『Query OK,』 と表示されれば削除完了。

```
mysql> select * from item;
```

③ 『select * from item;』 と入力。
select文でdelete文の内容が反映されたかを確認。

item_id	item_name	item_team	price
2	ぶどう	fall	200.00
3	さつまいも	fall	250.00
4	ネギ	winter	190.00
5	にんじん	spring	50.00
6	トマト	summer	100.00

5 rows in set (0.00 sec)

④ item_idが「1」のデータが削除されている。

delete文でデータの削除

```
mysql> delete from item;  
Query OK, 5 rows affected (0.00 sec)
```

① 『delete from item;』 と入力。

itemというテーブルのデータを全て削除するという意味。

② 『Query OK,』 と表示されれば削除完了。

```
mysql> select * from item;  
Empty set (0.00 sec)
```

③ 『select * from item;』 と入力。
select文でdelete文の内容が反映されたかを確認。

④ データが全て削除されているので、
『empty set』 と表示された。

drop文でテーブルやデータベース自体を削除

itemテーブル自体を**drop文で削除**

lesson02データベース自体を**drop文で削除**

drop文でテーブル自体の削除

```
mysql> show tables;
```

```
+-----+  
| Tables_in_lesson02 |  
+-----+  
| item                |  
+-----+  
1 row in set (0.00 sec)
```

```
mysql> drop table item;
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> show tables;
```

```
Empty set (0.00 sec)
```

① 『show tables;』と入力。

作成済みのテーブルを全て表示するという意味。
※今回は、「item」というテーブルしか作成していないので、1つしか表示れない。

② 『drop table item;』と入力。

テーブル自体を削除するという意味。

③ 『Query OK,』と表示されれば削除完了。

④ 『show tables;』と入力。
再度作成済みのテーブルを表示。

④ データが全て削除されているので、
『empty set』と表示された。

drop文でデータベース自体の削除

```
mysql> show databases;
```

① 『show databases;』 と入力。

作成済みのデータベースを全て表示するという意味。

```
+-----+  
| information_schema |  
| akira              |  
| lesson01           |  
| lesson02           |  
| mysql              |  
| performance_schema |  
| sys                |  
| umile              |  
+-----+
```

② lesson02のデータベースがあることを確認

```
8 rows in set (0.01 sec)
```

drop文でデータベース自体の削除

```
mysql> drop database lesson02;  
Query OK, 0 rows affected (0.01 sec)
```

① 『drop database lesson02;』 と入力。
lesson02のデータベースを削除するという意味。

```
mysql> show databases;
```

② 『Query OK,』 と表示されれば削除完了。

```
+-----+  
| Database |  
+-----+  
| information_schema |  
| akira |  
| lesson01 |  
| mysql |  
| performance_schema |  
| sys |  
| umile |  
+-----+  
7 rows in set (0.00 sec)
```

③ 『show databases;』 と入力。
再度作成済みのデータベースを表示。

④ lesson02というデータベースが表示されていない。
(=削除されたということがわかる)