

## セクション 4-ハンズオンスクリプト

### ■ テーブルに新しいデータを追加する

追加の前に、現在データベースとテーブルが用意されていない状態なので、

復習がてらに、データベースとテーブルをさくっと作っていきましょう。

```
CREAT DATABASE sales;
```

これで、データベースの作成が完了です。

次にデータベース「sales」に users テーブルを作っていきます。

カラムは下記の三つです。

id:—int（null 禁止、自動採番、主キーの設定）

name:—20char limit

age:—int 型

```
CREAT TABLE users (id int not null auto_increment primary key, name varchar(30),  
age int);
```

それでは、単数行のデータを追加していきます。

```
INSERT INTO users (id, name, age) VALUES (1,'takahashi', 31);
```

```
SELECT * FROM users;
```

こちらのコマンドで中身が正常に登録されているかを確認していきましょう。

※ちなみに SELECT 文については後の授業で詳しく解説いたしますので、一旦はテーブルからデータを取得するための命令ということだけ把握しておいてください。

すると正常に登録されていることが確認できます。

次に、複数行のレコードを追加してみましょう。

ここでは、カラム名を省略して実行してみましょう。

```
INSERT INTO users  
VALUE (2,'Suzuki', 37),(3,'sato', 21);
```

実行してみます。

正常に実行できました。

```
SELECT * FROM users;
```

を確認してみると、複数行追加されていることがわかります。

## ■データ追加 解説

Use 文でデータベースを指定します。

データベースを指定した上で、テーブルを作成します。

テーブル名は item とします。

カラムは

id,item\_name,price の三つとします。

コマンドは下記になります。

```
CREATE TABLE item(id int not null auto_increment primary key,  
item_name varchar(30), price int);
```

テーブルを作成したら、そのテーブルに内容をインサートしていきます。

```
INSERT INTO item VALUE
```

```
(1,'TV',100000),
```

```
(2,'smartphone',80000),
```

```
(3,'camera',50000);
```

正常に中身が登録されているかを確認するために以下のコマンドを打って、内容を確認します。

```
SELECT * FROM item;
```

すると正常に登録できていることが確認できると思います。

## ■データ更新の方法

まずは、テーブルに新しいカラム（購入頻度を示す） frequency を追加します。

コマンドは以下になります。

```
ALTER TABLE users ADD frequency VARCHAR(10) NOT NULL;
```

テーブル名は users で frequency という名前を指定して実行します。

念のために

```
DESC users;
```

でテーブル構造を確認しておきましょう。

それでは一括更新をしていきましょう。下記のコマンドを実行します。

```
UPDATE users SET frequency = 'still';
```

と入力して実行してみましょう。正常に実行できました。

```
SELECT * FROM users;
```

で確認しておきましょう。

すると、全レコードの「frequency」カラムに対して、 'still' 'の更新がかかったことを確認できます。

次に、条件指定をしてデータ更新をしていきましょう。

具体的にいえば、先ほど追加したデータ

```
id:1  
name: takahashi  
age: 31
```

に対して

年齢を 32 歳に変更しましょう

```
UPDATE users SET age = 32 WHERE id = 1;
```

と入力して実行してみましょう。

正常に実行できたと思います。

念のために、

```
SELECT * FROM users WHERE id = 1;
```

で確認しておきましょう。

すると 32 歳へ更新されたと思います。



## ■データ更新 解説

```
SELECT * from users;
```

でまずは、テーブルを確認しておきましょう。

更新前の状態を確認できたら、

指定された条件で実際に実行してみましょう。

```
UPDATE users SET frequency = repeater WHERE id >= 1;
```

これで、正常に実行できたことを確認できました。

```
SELECT * from users;
```

で再度確認してみると、

正常に' repeater' に更新できていることがわかります。

## ■データ削除

まずは、復習も込めて、UPDATE 文で id が 2 の鈴木さんに対して、「少ない」と更新していきましょう。

```
UPDATE users SET frequency = 'infrequent' WHERE id = 2;
```

これで鈴木さんの購入頻度が更新されました。念のために確認しておきます。

```
SELECT * users;
```

正常に更新されていることが確認できました。

それでは、（購入頻度を示す）frequency が「少ない」顧客のデータを削除していきましょう。

```
DELETE FROM users WHERE frequency = '少ない';
```

で実行してみましょう。

```
SELECT * users;
```

してみると、条件に合致した内容が削除されていることがわかります。

## ■テーブルデータ削除の課題 解説

まずは SELECT コマンドでテーブルの中身を事前に確認しておきます。

対象のテーブルは users テーブルです。

```
SELECT * FROM users;
```

すると現在は二つのレコードが存在することがわかります。

ここから delete 文を使用して、id1 の takahashiさんを削除していきます。

コマンドは以下の通りです。

```
DELETE FROM users WHERE id = 1;
```

SELECT 文で確認すると、削除されていることが確認できます。

ここから INSERT 文を使って一レコードを追加します。

今回は自動採番されているかどうかを確認するため id をあえて省略します。

```
INSERT INTO users (name,age,frequency) VALUES( 'kubota' ,24,' still' );
```

実行します。すると id 番号を指定していないですが、kubota さんの id が 4 となっています。

すなわち自動採番が引き継がれています。これで、課題の一つ目の解説がお話ありました。

次に TRUNCATE 文を使って、一括削除した後に、複数レコードを追加して、自動採番がリセットされているかを確認します。

```
TRUNCATE TABLE users;
```

これでテーブルの中身が削除されました。

ここから複数レコードを登録します。

```
INSERT INTO users (name,age,frequency) VALUES  
( 'takahashi' ,31,' still' ),  
( 'sato' ,21,' still' );
```

中身を見てみると、自動採番がリセットされていることがわかります。