

## 4-4-8.DELETE文

### はじめに

DELETE文は、SQLのなかでもデータ操作言語と呼ばれ、データベース内のデータを削除する命令です。DELETE文を使用することによって、テーブル内の不要になったデータを削除することができます。一方でDELETE文を使わない削除命令もあります。

### Step1: 概念を知る

削除は2種類あります。

#### 物理削除

データベース上からデータを削除します。  
特別な手続きをすれば削除したデータを復元することができますが、  
基本的にはデータを完全に削除することを前提に行います。

```
DELETE FROM "テーブル名"
```

#### 論理削除

データベース上からデータを見かけ上削除します。  
データに削除されたという目印（削除フラグ）を設定し、  
見かけ上削除されているようにするだけで、実際には削除されていません。  
削除フラグを立てたデータを扱うこともできますし、復元も容易に行うことができます。

```
UPDATE "テーブル名"  
SET "カラム名" = TRUE (or FALSE)
```

もしくは、'0（生存）'、'1（削除）'を設定したり、  
削除フラグではなく、データステータス（DATA\_STS）などとして、  
以下のようにアルファベットを付与したりして管理します。

リテラル	意味
'E'	生存（Exists）
'D'	削除済み（Delete）
'P'	保留中（Pending）

### Step2: 使い方を知る

では、さっそくテーブル内のデータを削除してみましょう。

#### サンプル

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

postgres on postgres@PostgreSQL 10

```

1 DELETE FROM person
2 WHERE person_id = 1

```

データ出力 EXPLAIN メッセージ クエリの履歴

DELETE 1

クエリが 69 msec で成功しました

postgres on postgres@PostgreSQL 10

```

1 SELECT *
2 FROM person

```

データ出力 EXPLAIN メッセージ クエリの履歴

	person_id integer	fname character varying (20)	lname character varying (20)	gender character (1)	birth_date date
1	2	Jonny	Buckland	M	1977-09-11
2	3	Guy	Berryman	M	1978-04-12
3	4	Will	Champion	M	1978-07-31

## 解説

1. `DELETE FROM` を宣言し、テーブル名 `person` をつけます。
2. 削除したい値、行が決まっている場合は `WHERE句` で指定します。

実行して構文にエラーがなければテーブル内のデータが変更されます。

これで、不要になったデータを削除してテーブルを新しくすることができます。

DELETE文もUPDATE文同様、

`WHERE句` で条件を指定しなければテーブルすべてに対して削除命令を実行します。

postgres on postgres@PostgreSQL 10

```
1 DELETE FROM person
```

データ出力 EXPLAIN メッセージ クエリの履歴

DELETE 4

クエリが 96 msec で成功しました

postgres on postgres@PostgreSQL 10

```

1 SELECT *
2 FROM person

```

データ出力 EXPLAIN メッセージ クエリの履歴

	person_id integer	fname character varying (20)	lname character varying (20)	gender character (1)	birth_date date
1	2	Jonny	Buckland	M	1977-09-11

全てのデータを削除する分には問題ありませんが、  
こちらも **WHERE句** を指定するクセをつけましょう。  
ちなみに、テーブル内のデータを全て削除しても、テーブル自体は残ります。

## 使い分け

### 物理削除

#### メリット

- 1: delete文の宣言で簡単にできる。
- 2: データが減るため、データベースのパフォーマンスが上がる。

#### デメリット

- 1: データの復元が難しい。
- 2: 削除したら参照不可能になる。

### 論理削除

#### メリット

- 1: データの削除、復元が容易。
- 2: 削除済みデータの参照が可能。

#### デメリット

- 1: テーブルに削除フラグ用の列追加が必要。
- 2: データを参照する際、削除フラグを立てたデータを表示しないようにクエリを作成する必要がある。
- 3: データが増えるため、データベースのパフォーマンスが下がる。

データがもう必要ないとわかっている際には、物理削除を行い、  
削除したデータを見ながら、データベースを管理したい際には、論理削除を行う  
といった使い分けが出来るようになるといいでしよう。

## 課題

提出課題はありませんので、一通り学習が終わったら次の章に進んで下さい。

最終更新日時: 2022年 09月 10日(土曜日) 07:49