

## 4-4-6.INSERT文

# INSERT文

### はじめに

INSERT文は、SQLのなかでもデータ操作言語と呼ばれ、データベース内にデータを新規登録する命令です。INSERT文を使用することによって、テーブルにデータを持たせることができるようになります。

### Step1: 概念を知る

**INSERT文**  
テーブルにデータを挿入する際に用います。

```
INSERT INTO "テーブル名"  
(列名1, 列名2, ... )  
VALUES  
(値1, 値2, ... )
```

### Step2: 使い方を知る

では、さっそくテーブルにデータを挿入してみましょう。

postgres on postgres@PostgreSQL 10

1

INSERT INTO person

2

(person\_id, fname, lname, gender, birth\_date)

3

VALUES

4

(1, 'Chris', 'Martin', 'M', '1977/03/02');

5

6

INSERT INTO person

7

(person\_id, fname, lname, gender, birth\_date)

8

VALUES

9

(2, 'Jonny', 'Buckland', 'M', '1977/09/11');

10

11

INSERT INTO person

12

(person\_id, fname, lname, gender, birth\_date)

13

VALUES

14

(3, 'Guy', 'Berryman', 'M', '1978/04/12');

15

16

INSERT INTO person

17

(person\_id, fname, lname, gender, birth\_date)

18

VALUES

19

(4, 'Will', 'Champion', 'M', '1978/07/31');

データ出力

EXPLAIN

メッセージ

クエリの履歴

INSERT 0 1

クエリが 72 msec で成功しました

解説

- 1. INSERT INTO を宣言し、テーブル名 person をつけます。
- 2. カラム名を ( ) 内に列挙します。
- 3. カラム名に対応する値を順に、テーブル定義に合わせて ( ) 内に列挙します。



実行して構文にエラーがなければテーブルにデータが挿入されます。  
これで、データを体系的に扱うことができます。

補足

実際にデータがテーブルに挿入されたかどうかは、**SELECT文** を使って確認することができます。

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	1	Chris	Martin	M	1977-03-02
2	2	Jonny	Buckland	M	1977-09-11
3	3	Guy	Berryman	M	1978-04-12
4	4	Will	Champion	M	1978-07-31

こういう場合はエラー

#### 1: NOT NULLの列に値を挿入しなかった場合

**NOT NULL** が指定されている列は空欄にできませんので、値の挿入が必須になります。

#### 2: 列数と挿入する値の数が一致していない場合

列数が5あるのに、値が4しかないとクエリを実行できません。

ただし、列に **NOT NULL** を指定していなければ空白にすることもできます。

その際は **NULL** が登録されます。

#### 3: 列に指定している型と、挿入する値の型が一致していない場合

**INT型** の列に文字列を挿入、**VARCHAR型**の列に挿入する文字列に '' をつけないなど、型と挿入する値が一致していないとクエリの実行ができません。

## 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) 07:45