

4-4-3.CREATE文

はじめに

CREATE文は、SQLのなかでもデータ定義言語と呼ばれ、データベースそのものを操作する命令です。特に使用する機会が多いのが、データベース内にテーブルを作成する際に使う **CREATE TABLE文** です。データを入れるハコになるので、テーブルの設計次第でデータの使い勝手が大きく変わります。

Step1: 概念を知る

・ **CREATE TABLE文**

新しいテーブルを作成する際に用います。
テーブル名と、列の名前、型を指定することで、新しいテーブルを作成することができます。

```
CREATE TABLE "テーブル名" (  
  カラム定義1,  
  カラム定義2,  
  カラム定義3,  
  ...  
)
```

Step2: 使い方を知る

今回は人の情報を管理する、personというテーブルを作成してみます。
personテーブルはこのような定義にします。

【テーブル名 : person】

カラム名	型	カラムの説明	備考
person_id	INT	ID	主キー
fname	VARCHAR(20)	名	NOT NULL
lname	VARCHAR(20)	氏	NOT NULL
gender	CHAR(1)	性別	
birth_date	DATE	生年月日	

解説

決まった形があるわけではありませんが、上記のような表は、テーブル設計の際に使われることが多い書き方です。

・ **カラム名**

列のタイトルです。
自由につけることができますが、カラム名をもとにデータを抽出（この後の章で学習）しますので、分かりやすい名前を付けた方がいいでしょう。
単語同士は **_**（アンダースコア）でつなぐのが通例です。

・ **型**

これまでの学習でも出てきましたので、何となくイメージがつくのではないのでしょうか。
SQLにも入力する値によって型が決まっています。

型の例	型名
数値の場合	INT など
少数の場合	FLOAT など
文字列の場合	CHAR、VARCHAR、TEXT など
日付の場合	DATE など

補足

CHAR、VARCHAR は文字列データを扱えることは同じですが、
CHAR は **固定長**、VARCHAR は **可変長** といい、格納されるデータによって挙動が変わります。

列定義をそれぞれ CHAR(10)、VARCHAR(10) として、
abc というデータを格納した場合は以下のようになります。

・ CHAR の場合

文字列 abc の右側に 7バイト分の空白が自動で挿入され、10バイトに固定 されます。

・ VARCHAR の場合

文字列abc は 3バイトのまま格納されるよう動的に調整 されます。

格納するデータが郵便番号など、**長さが一定のもの** の場合は CHAR、
人の名前など、**長さが一定でないもの** の場合は、 VARCHAR などと使い分けるといいでしょう。

また、固定長である場合、「登録したデータが定義した長さに満たない可能性があるかどうか」を考慮した設計 でなくてはなりません。

例えば、WHERE句の条件として ID = 'abc' とした場合に、
'abc ' (←気づきづらいですが、空白埋めで10文字) のデータは検索結果にひっかかりません。

こういったDB設計とプログラマーの実装とが噛み合っていない場合、
この問題自体に気づきづらいため、この項目は本当に CHAR でいいのか？という点はしっかりと考慮しなくてはなりません。

・ 備考

主キー や NOT NULL など、テーブルに設定した定義をここに記述します。

1. 主キーとは、一意で重複がないキー です。
同じテーブル内で重複するデータを持つことはできません。
また、主キーに指定した列は自動的に NOT NULL になります。
2. NOT NULL とは、空欄を許容しない という意味です。
値を入力しないとエラーになります。
必ず値を入力してほしいところに設定するといいでしょう。

では、テーブルを作成しましょう！

```
postgres on postgres@PostgreSQL 10
1 CREATE TABLE person (
2   person_id INT PRIMARY KEY,
3   fname VARCHAR(20) NOT NULL,
4   lname VARCHAR(20) NOT NULL,
5   gender CHAR(1),
6   birth_date DATE
7 )
```

データ出力 EXPLAIN メッセージ クエリの履歴

CREATE TABLE

クエリが 133 msec で成功しました

解説

- 1. CREATE TABLE を宣言し、テーブル名 person をつけます。
- 2. テーブルの情報は () 内に記述します。
- 3. カラム名 fname、型 VARCHAR、入力可能数 (20) の順に記述します。
※中には DATE型 のように、決まった出力形式があり、入力可能数の記述が不要な型もあります。

実行して構文にエラーがなければテーブルが作成されます。
これで、データを入れることができるようになります。

補足

テーブルを作成しても何か結果が返されるわけではないので、本当に作成できたか不安に思うかもしれません。
そこで、テーブルの定義を確認することによって、実際にテーブルが作成されたか確認することが出来ます。
まだ学習していない構文を多く含んでいますので、参考程度に見るだけで大丈夫です。

書き方

```
SELECT *
FROM information_schema.columns
WHERE table_name = '確認したいテーブル名'
ORDER BY ordinal_position;
```

※ ORDER BY は付与しなくても問題ありません

postgres on postgres@PostgreSQL 10

```
1 SELECT *
2 FROM information_schema.columns
3 WHERE table_name = 'person'
4 ORDER BY ordinal_position;
```

データ出力 EXPLAIN メッセージ クエリの履歴

table_name	column_name	ordinal_position	is_nullable	data_type
character varying	character varying	integer	character varying (3)	character varying
person	person_id	1	NO	integer
person	fname	2	NO	character varying
person	lname	3	NO	character varying
person	gender	4	YES	character
person	birth_date	5	YES	date

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) 07:19