枚举类型是包含一系列有序的静态值集合的一个数据类型。

**枚举类型的使用**

在PostgreSQL中要使用枚举类型需要先使用CREATE TYPE来创建此枚举类型。 示例如下。

先建一个名为“mood”的枚举类型， 并建一张测试表：

CREATE TYPE mood AS ENUM ('sad', 'ok', 'happy');

CREATE TABLE person (name text,current\_mood mood);

INSERT INTO person VALUES (‘zhangsan', ‘sad');

INSERT INTO person VALUES (‘lisi', ‘ok');

INSERT INTO person VALUES (‘wangwu', 'happy');

INSERT INTO person VALUES (‘zhaoliu', ‘not ok');--输入的字符串不在枚举类型中，则报错

SELECT \* FROM person WHERE current\_mood = 'happy';

SELECT \* FROM person WHERE current\_mood = ‘not ok'; --输入的字符串不在枚举类型中，则报错

查看枚举类型的定义：

\dT+ mod

select \* from pg\_enum;

**枚举类型说明**

在枚举类型中， 值的顺序是创建枚举类型时定义的顺序。 所有的比较标准运算符及相关的聚集函数都可支持枚举类型。 示例如下：

SELECT min(current\_mood), max(current\_mood) FROM person;

SELECT \* FROM person where current\_mood = (SELECT max(current\_mood) FROM person);

注意：枚举类型的值对大小写是敏感的， 如“sad”不等于“Sad”。 标签中的空格也是一样，如“sad ”（“sad”后有一个空格） 不等于“sad”。

**枚举类型的函数**

CREATE TYPE rainbow AS ENUM ('red', 'orange', 'yellow','green', 'blue', 'purple');

|  |  |  |
| --- | --- | --- |
| 函数 | 描述 | 例子 |
| enum\_first | 返回枚举类型的第一个值 | enum\_first(null::rainbow) → red  enum\_first(‘red’::rainbow) → red |
| enum\_last | 返回枚举类型的最后一个值 | enum\_last(null::rainbow) → purple |
| enum\_range | 返回枚举类型的所有值 | enum\_range(null::rainbow) → {red,orange,yellow,​green,blue,purple} |
| enum\_range | 返回两个值之间的范围 | enum\_range('orange'::rainbow, 'green'::rainbow) → {orange,yellow,green}  enum\_range(NULL, 'green'::rainbow) → {red,orange,​yellow,green}  enum\_range('orange'::rainbow, NULL) → {orange,yellow,green,​blue,purple} |

除了两个参数形式的enum\_range外， 其余函数会忽略传递给它们的具体值， 因为它们只关心声明的数据类型。 使用null加上类型转换也会得到相同的结果， 示例如下：

select enum\_first(null::week), enum\_last(null::week);