第三次平时作业

第1题

1. 编写一个函数，函数名为get_student_phone，无接收参数，返回一个随机的手机号，长度11位，手机号以'159'或 '137'开头，要求任意满足该要求的手机号能等概率生成。

   示例：

```
select get_student_phone();
get_student_phone
-------------------
13790119007
```

第1题

- 答题

```
CREATE or replace function get_student_phone() returns bigint as $$
DECLARE
    latter bigint ;
    former bigint;
BEGIN
    latter := trunc(random()*100000000);
    if random() > 0.5 then
        former := 13700000000;
    else
        former := 15900000000;
    end if;

    RETURN former+latter;
END;
$$ language plpgsql;
```

第2题

2. 编写一个函数，函数名为get_student_date，无接收参数，返回一个随机的日期，日期格式为'YYYY-MM-DD'。要求返回的日期区间为[2020-01-01, 2021-12-31]，其中，要求生成2020年份概率为60%，生成2021年份概率为40%，此外，月和日则是等概率返回。

   示例：

```
select get_student_date();
get_student_date
---------------
2020-12-13
```

第2题

- 答题

```
CREATE or replace function get_student_date() returns date as $$
DECLARE
    days int;
    dates date;
    first_date date;
BEGIN
    first_date := '2020-01-01';
    if random() > 0.4  then
        days := trunc( random()*366 );
        dates := first_date + days;
    else
        days := trunc( random()*365 );
        dates := first_date + 366 + days;
    end if;

    RETURN dates;
END;
$$ language plpgsql;
```

第3题

3. 编写一个函数，函数名为create_student_table，无接收参数。在该函数中，新建一个数据表 student，该数据表拥有3个字段，分别是 student_id, phone_num, enrollment_date，其中 student_id 为自增的序列，从1开始自增，且为主键；然后，往该数据表新增 15条记录，这 15条记录中，phone_num和 enrollment_date 分别使用上述自己编写的第一个和第二个函数生成。最后返回该表。该函数理应可以连续调用多次，每次生成并返回的表都不一样。

   示例：

```
select * from create_student_table();
student_id | phone_num    | enrollment_date
------------------------------------
    1      | 13790829207  | 2020-02-25
...
    15     | 15923624934  | 2021-04-15
```

第3题

- 答题

```
CREATE or replace function create_student_table() returns table(student_id int , phone_num
bigint , enrollment_date date) as $$
    DECLARE
    BEGIN
        drop table if exists student;
        create table student(student_id int not null, phone_num bigint not null,
enrollment_date date not null, primary key(student_id) );

        for i in 1..15 loop
            insert into student values(i , get_student_phone() , get_student_date());
        end loop;
        return query select* from student;
END;
$$ language plpgsql;
```

第4题

4. 查询：使用 student表，找出所有enrollment_date在2020年7月1日（包括这一天）之后的学生，并输出其phone_num。

第4题

- 答题

```
select phone_num
from student
where enrollment_date >= '2020-07-01';
```