数据库第三次作业

米家龙 18342075

1. 编写一个函数，函数名为 `get_rand_phone_num` ，无接收参数，返回一 个随机的电话号码，长度11位，电话号码以'132'或 '137'或 '189'或 '135'开头，要求任意满足该要求的电话号码能等概率生成。

```
db=# CREATE OR REPLACE FUNCTION get_rand_phone_num()
db-# RETURNS TEXT AS $$
db$#   DECLARE
db$#     head INTEGER[] = '{132, 135, 137, 189}';
db$#     tmp INTEGER;
db$#     ans INTEGER;
db$#     ans_t TEXT;
db$#   BEGIN
db$#     tmp := TRUNC(RANDOM() * 4 + 1)::INT; -- 获取头
db$#     ans_t := head[tmp]::TEXT;
db$#     tmp := 0;
db$#     WHILE tmp < 8 LOOP
db$#       ans := TRUNC(RANDOM() * 10)::INT;
db$#       ans_t := ans_t || ans::TEXT;
db$#       tmp = tmp + 1;
db$#     END LOOP;
db$#     RETURN ans_t;
db$#   END
db$# $$ LANGUAGE PLPGSQL STRICT;
CREATE FUNCTION
db=# SELECT get_rand_phone_num();
 get_rand_phone_num
--------------------
 13229822893
(1 row)
```

2. 编写一个函数，函数名为 get_rand_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[1990-01-01, 2000-01-01)，所有满足要求的日期能被等概率返回。

```
db=# CREATE OR REPLACE FUNCTION get_rand_date()
db-# RETURNS DATE as $$
db$#   DECLARE
db$#     down_limit DATE = '1990-01-01';
db$#     up_limit DATE = '2000-01-01';
db$#     range INT = up_limit - down_limit;
db$#     tmp INT;
db$#   BEGIN
db$#     tmp := TRUNC(RANDOM() * range)::INT;
db$#     RETURN down_limit + tmp;
db$#   END;
db$# $$ LANGUAGE PLPGSQL STRICT;
CREATE FUNCTION
db=# SELECT get_rand_date();
 get_rand_date
---------------
 1992-05-05
(1 row)

db=# SELECT get_rand_date();
 get_rand_date
---------------
 1993-11-10
(1 row)
```

3. 编写一个函数，函数名为 create_table_test1，无接收参数。在该函数 中，新建一个数据表test1，该数据表拥有3个字段，分别是id, rand_phone, rand_date，其中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10条记录，这10条记录中，rand_phone 和rand_date使用上述自己编写的函数随机生成。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。

```
db=# CREATE OR REPLACE FUNCTION create_table_test1()
db-# RETURNS SETOF test1 AS $$
db$# DECLARE
db$#   num INT = 1;
db$#   res RECORD;
db$# BEGIN
db$#   IF (TO_REGCLASS('test1') IS NOT NULL) THEN
db$#     TRUNCATE TABLE test1;
db$#   ELSE
db$#     CREATE TABLE test1 (
db$#       id SERIAL PRIMARY KEY,
db$#       rand_phone TEXT,
db$#       rand_date DATE
db$#     );
db$#   END IF;
db$#
db$#   WHILE (num <= 10) LOOP
db$#     INSERT INTO test1 VALUES (num, get_rand_phone_num(), get_rand_date());
db$#     num = num + 1;
db$#   END LOOP;
db$#   FOR res IN SELECT * FROM test1 LOOP
db$#     RETURN NEXT res;
db$#   END LOOP;
db$#   RETURN;
db$# END;
db$# $$ LANGUAGE PLPGSQL STRICT;
CREATE FUNCTION
db=#
db=# -- 测试用
db=# SELECT * FROM create_table_test1();
 id | rand_phone  | rand_date
----+-------------+------------
  1 | 13733616549 | 1997-10-21
  2 | 13270316421 | 1994-02-18
  3 | 13544742036 | 1991-01-03
  4 | 13534806451 | 1994-05-22
  5 | 13521453005 | 1991-12-03
  6 | 13506970174 | 1998-11-02
  7 | 18982372918 | 1990-07-03
  8 | 13722768069 | 1996-06-24
  9 | 18946534652 | 1990-07-20
 10 | 13519746048 | 1998-12-06
(10 rows)

db=#
```

4. 编写一个函数，函数名为 create_table_test2，无接收参数。在该函数 中，新建一个数据表test2，该数据表拥有2个字段，分别是id和info， 其中id类型为整数，同时为主键，info为json格式数据；然后，把 test1中的所有记录首先转换为如下的json格式： {rand_phone:'rand_pnhone', rand_date:'rand_date'}，即拥有2个键值 对，键分别是rand_phone和rand_date，值为test1表中对应的值；最 后，把test1的数据生成json格式的记录插入test2中。test2的id与 test1对应的id相同。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和test1一样，即不重复调用create_table_test1的 情况下，多次调用create_table_test2返回的表应该是相同的。

```
db=# SELECT * FROM test1
db-# ;
 id | rand_phone  | rand_date
----+-------------+------------
  1 | 13717617835 | 1994-03-27
  2 | 18950848479 | 1990-09-25
  3 | 13710944974 | 1991-07-19
  4 | 13512173867 | 1994-02-28
  5 | 13279063911 | 1998-08-11
  6 | 13270274662 | 1991-12-19
  7 | 18963071457 | 1992-08-25
  8 | 13270977378 | 1994-09-07
  9 | 13702639195 | 1993-07-04
 10 | 13249557693 | 1993-05-26
(10 rows)

db=# SELECT * FROM create_table_test2();
 id |                              info
----+-----------------------------------------------------------
  1 | {"rand_phone" : "13717617835", "rand_date" : "1994-03-27"}
  2 | {"rand_phone" : "18950848479", "rand_date" : "1990-09-25"}
  3 | {"rand_phone" : "13710944974", "rand_date" : "1991-07-19"}
  4 | {"rand_phone" : "13512173867", "rand_date" : "1994-02-28"}
  5 | {"rand_phone" : "13279063911", "rand_date" : "1998-08-11"}
  6 | {"rand_phone" : "13270274662", "rand_date" : "1991-12-19"}
  7 | {"rand_phone" : "18963071457", "rand_date" : "1992-08-25"}
  8 | {"rand_phone" : "13270977378", "rand_date" : "1994-09-07"}
  9 | {"rand_phone" : "13702639195", "rand_date" : "1993-07-04"}
 10 | {"rand_phone" : "13249557693", "rand_date" : "1993-05-26"}
(10 rows)

db=# SELECT * FROM create_table_test2();
 id |                              info
----+-----------------------------------------------------------
  1 | {"rand_phone" : "13717617835", "rand_date" : "1994-03-27"}
  2 | {"rand_phone" : "18950848479", "rand_date" : "1990-09-25"}
  3 | {"rand_phone" : "13710944974", "rand_date" : "1991-07-19"}
  4 | {"rand_phone" : "13512173867", "rand_date" : "1994-02-28"}
  5 | {"rand_phone" : "13279063911", "rand_date" : "1998-08-11"}
  6 | {"rand_phone" : "13270274662", "rand_date" : "1991-12-19"}
  7 | {"rand_phone" : "18963071457", "rand_date" : "1992-08-25"}
  8 | {"rand_phone" : "13270977378", "rand_date" : "1994-09-07"}
  9 | {"rand_phone" : "13702639195", "rand_date" : "1993-07-04"}
 10 | {"rand_phone" : "13249557693", "rand_date" : "1993-05-26"}
(10 rows)
```

5. 查询：使用test2表，找出所有日期在1999年3月20日（包括这一天）之后的记录的电话号码。

```
HINT:  No function matches the given name and argument types. You might need to add ex
db=# SELECT * FROM test2
db-# ;
 id |                          info
----+-------------------------------------------------------------
  1 | {"rand_phone" : "13717617835", "rand_date" : "1994-03-27"}
  2 | {"rand_phone" : "18950848479", "rand_date" : "1990-09-25"}
  3 | {"rand_phone" : "13710944974", "rand_date" : "1991-07-19"}
  4 | {"rand_phone" : "13512173867", "rand_date" : "1994-02-28"}
  5 | {"rand_phone" : "13279063911", "rand_date" : "1998-08-11"}
  6 | {"rand_phone" : "13270274662", "rand_date" : "1991-12-19"}
  7 | {"rand_phone" : "18963071457", "rand_date" : "1992-08-25"}
  8 | {"rand_phone" : "13270977378", "rand_date" : "1994-09-07"}
  9 | {"rand_phone" : "13702639195", "rand_date" : "1993-07-04"}
 10 | {"rand_phone" : "13249557693", "rand_date" : "1993-05-26"}
(10 rows)

db=# SELECT info->'rand_phone' FROM test2
  WHERE
    TO_DATE(info::JSON->>'rand_date', 'YYYY-MM-DD') > date '1999-03-20';
 ?column?
----------
(0 rows)
```