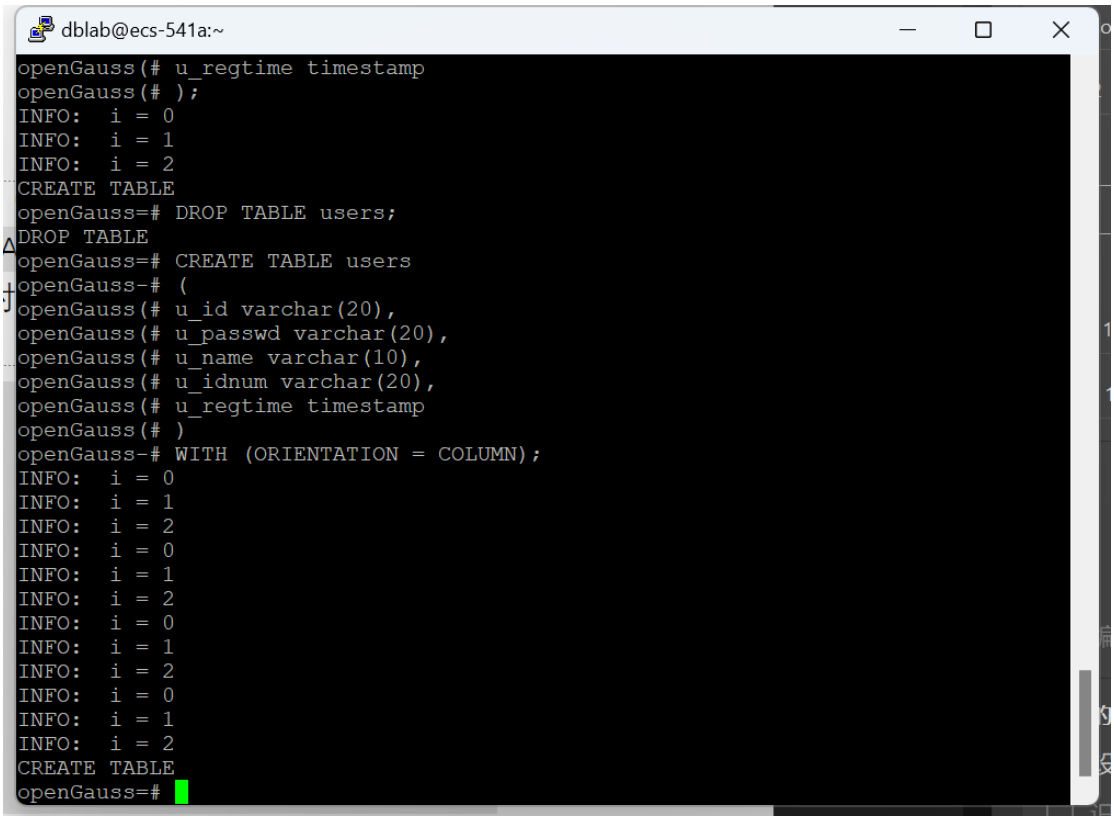
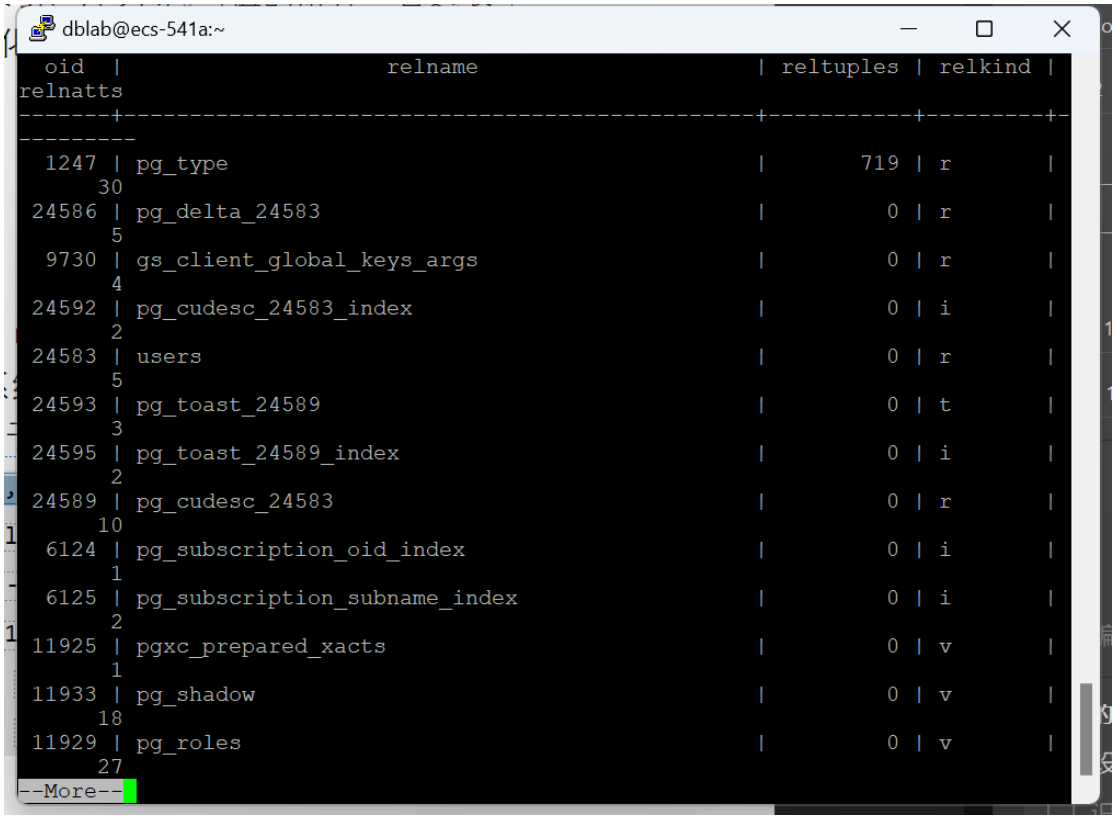
# 第 7 章 表的创建与系统表

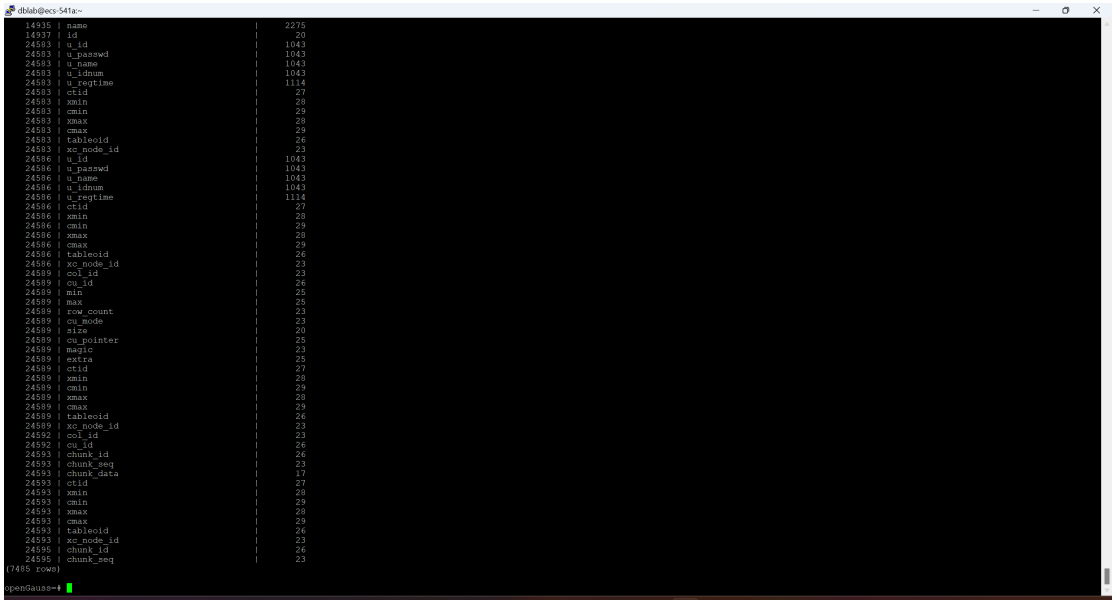
## 学号：3022005055 姓名：李燚坡 班级：人工智能4班

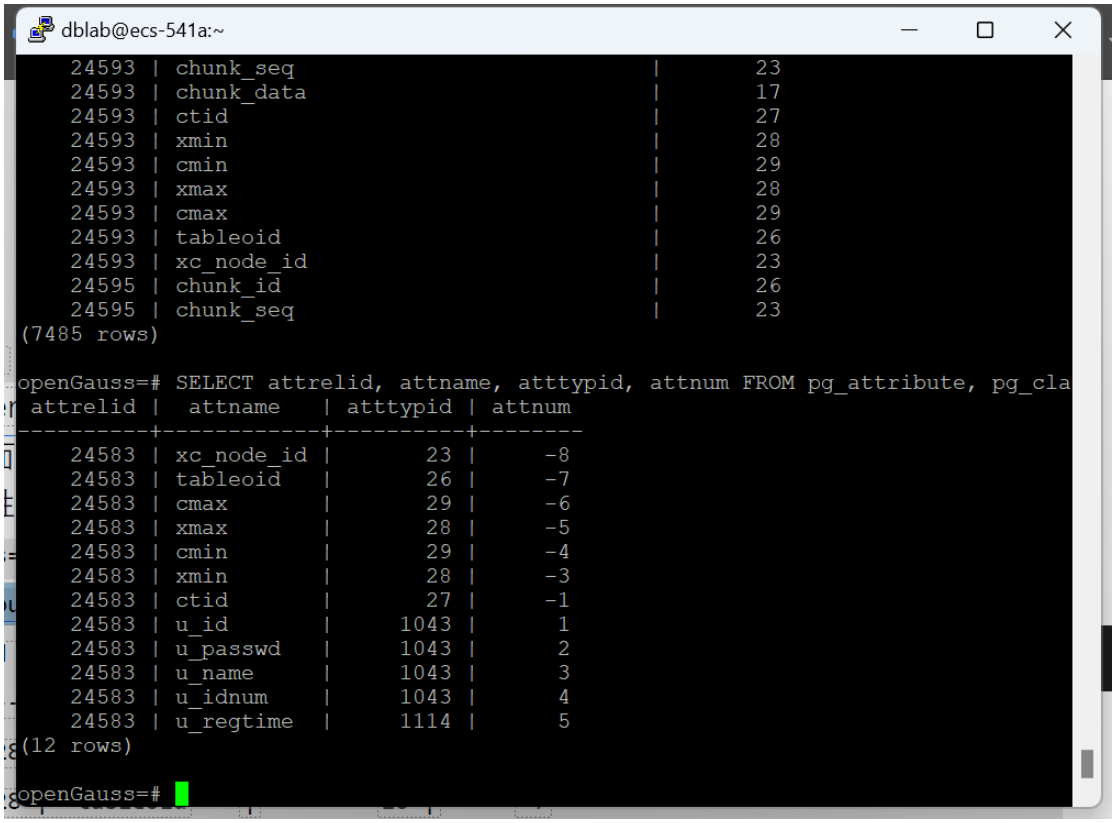
1. 完成实验步骤 7.4.1，创建 user 行存表和列存表。

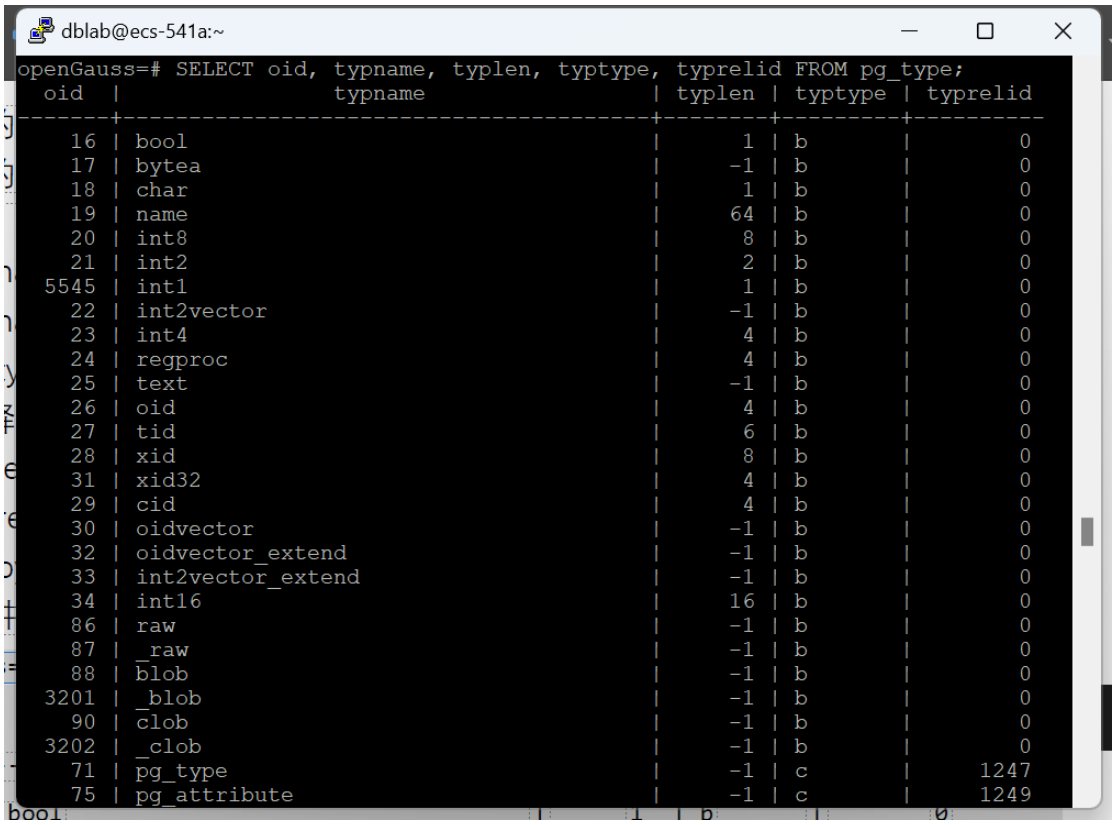


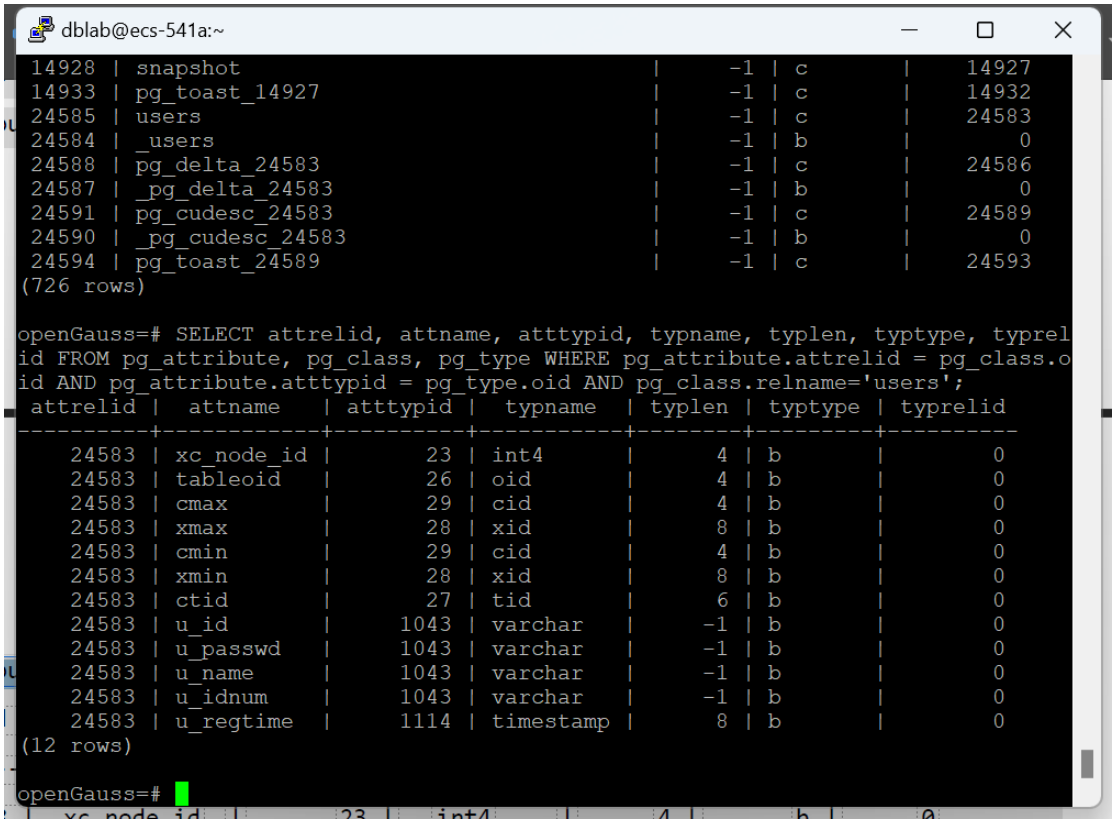

2. 2. 完成实验步骤 7.4.2，就其中给出的系统表属性，画出系统表 pg_class、pg_attribute、pg_type 之间的关系的 ER 图。


dblab@ecs-541a:~
oid | relname | reltuples | relkind | relnatts
1247 | pg_type | 719 | r | 30
24586 | pg_delta_24583 | 0 | r | 5
9730 | gs_client_global_keys_args | 0 | r | 4
24592 | pg_cudesc_24583_index | 0 | i | 2
24583 | users | 0 | r | 5
24593 | pg_toast_24589 | 0 | t | 3
24595 | pg_toast_24589_index | 0 | i | 2
24589 | pg_cudesc_24583 | 0 | r | 10
6124 | pg_subscription_oid_index | 0 | i | 1
6125 | pg_subscription_subname_index | 0 | i | 2
11925 | pgxc_prepared_xacts | 0 | v | 1
11933 | pg_shadow | 0 | v | 18
11929 | pg_roles | 0 | v | 27
--More--


dblab@ecs-541a:~
14935 | name | 2275
14937 | id | 20
24583 | u_id | 1043
24583 | u_passwd | 1043
24583 | u_name | 1043
24583 | u_idnum | 1043
24583 | u_regtime | 1114
24583 | ctid | 27
24583 | xmin | 28
24583 | cmin | 29
24583 | xmax | 28
24583 | cmax | 29
24583 | tableoid | 26
24583 | xc_node_id | 23
24586 | u_id | 1043
24586 | u_passwd | 1043
24586 | u_name | 1043
24586 | u_idnum | 1043
24586 | u_regtime | 1114
24586 | ctid | 27
24586 | xmin | 28
24586 | cmin | 29
24586 | xmax | 28
24586 | cmax | 29
24586 | tableoid | 26
24586 | xc_node_id | 23
24589 | col_id | 23
24589 | cu_id | 26
24589 | min | 25
24589 | max | 25
24589 | row_count | 23
24589 | cu_mode | 23
24589 | size | 20
24589 | cu_pointer | 25
24589 | magic | 23
24589 | extra | 25
24589 | ctid | 27
24589 | xmin | 28
24589 | cmin | 29
24589 | xmax | 28
24589 | cmax | 29
24589 | tableoid | 26
24589 | xc_node_id | 23
24592 | col_id | 23
24592 | cu_id | 26
24593 | chunk_id | 26
24593 | chunk_seq | 23
24593 | chunk_data | 17
24593 | ctid | 27
24593 | xmin | 28
24593 | cmin | 29
24593 | xmax | 28
24593 | cmax | 29
24593 | tableoid | 26
24593 | xc_node_id | 23
24595 | chunk_id | 26
24595 | chunk_seq | 23
(7485 rows)
openGauss=#

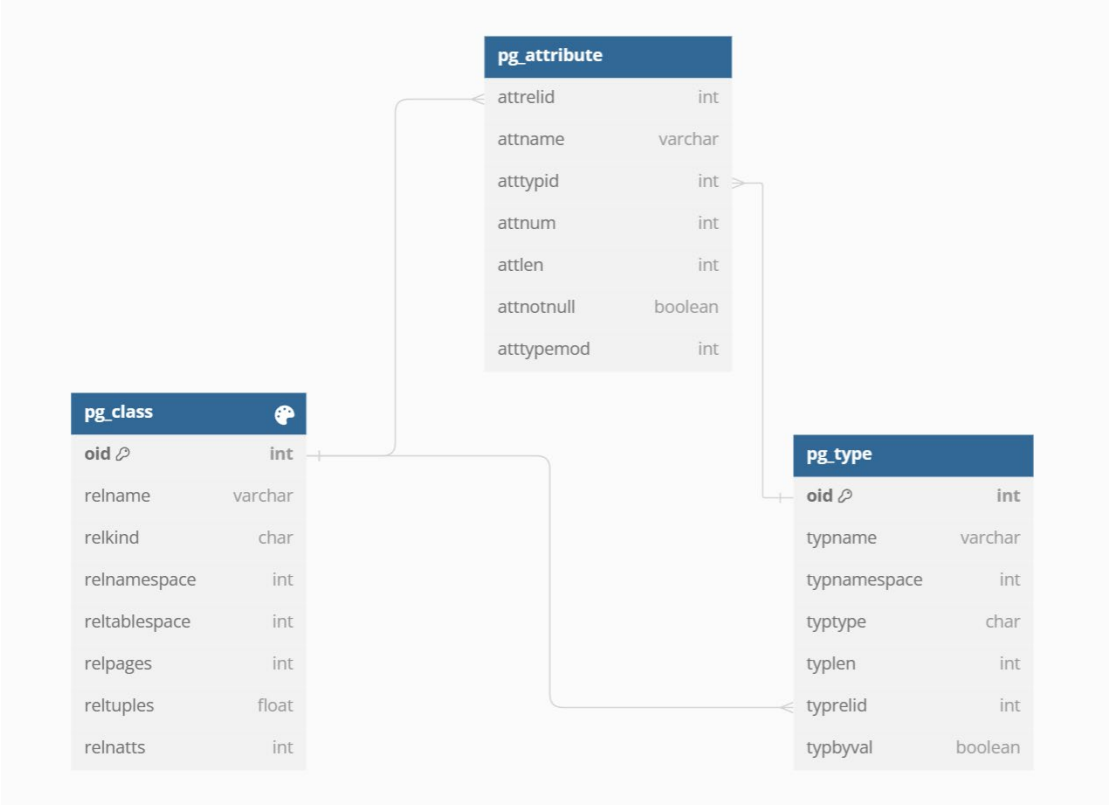

dblab@ecs-541a:~
24593 | chunk_seq | 23
24593 | chunk_data | 17
24593 | ctid | 27
24593 | xmin | 28
24593 | cmin | 29
24593 | xmax | 28
24593 | cmax | 29
24593 | tableoid | 26
24593 | xc_node_id | 23
24595 | chunk_id | 26
24595 | chunk_seq | 23
(7485 rows)
openGauss=# SELECT attrelid, attname, atttypid, attnum FROM pg_attribute, pg_cla
attrelid | attname | atttypid | attnum
24583 | xc_node_id | 23 | -8
24583 | tableoid | 26 | -7
24583 | cmax | 29 | -6
24583 | xmax | 28 | -5
24583 | cmin | 29 | -4
24583 | xmin | 28 | -3
24583 | ctid | 27 | -1
24583 | u_id | 1043 | 1
24583 | u_passwd | 1043 | 2
24583 | u_name | 1043 | 3
24583 | u_idnum | 1043 | 4
24583 | u_regtime | 1114 | 5
(12 rows)
openGauss=#


dblab@ecs-541a:~
openGauss=# SELECT oid, typname, typlen, typtype, typrelid FROM pg_type;
oid | typname | typlen | typtype | typrelid
16 | bool | 1 | b | 0
17 | bytea | -1 | b | 0
18 | char | 1 | b | 0
19 | name | 64 | b | 0
20 | int8 | 8 | b | 0
21 | int2 | 2 | b | 0
5545 | int1 | 1 | b | 0
22 | int2vector | -1 | b | 0
23 | int4 | 4 | b | 0
24 | regproc | 4 | b | 0
25 | text | -1 | b | 0
26 | oid | 4 | b | 0
27 | tid | 6 | b | 0
28 | xid | 8 | b | 0
31 | xid32 | 4 | b | 0
29 | cid | 4 | b | 0
30 | oidvector | -1 | b | 0
32 | oidvector_extend | -1 | b | 0
33 | int2vector_extend | -1 | b | 0
34 | int16 | 16 | b | 0
86 | raw | -1 | b | 0
87 | _raw | -1 | b | 0
88 | blob | -1 | b | 0
3201 | _blob | -1 | b | 0
90 | clob | -1 | b | 0
3202 | _clob | -1 | b | 0
71 | pg_type | -1 | c | 1247
75 | pg_attribute | -1 | c | 1249




```
openGauss=# SELECT typname, typlen, typtype, typrelid FROM pg_class, pg_type WHE
RE pg_class.oid = pg_type.typrelid AND pg_class.relname='users';
 typname | typlen | typtype | typrelid
---------+--------+---------+----------
 users   |     -1 | c       |    24583
(1 row)
```

```
openGauss=# SELECT oid, reltype FROM pg_class WHERE relname='users';
  oid  | reltype
-------+---------
 24583 |   24585
(1 row)
```


pg_attribute
attrelid int
attname varchar
atttypid int
attnum int
attlen int
attnotnull boolean
atttypemod int
pg_class
oid int
relname varchar
relkind char
relnamespace int
reltablespace int
relpages int
reltuples float
relnatts int
pg_type
oid int
typname varchar
typnamespace int
typtype char
typlen int
typrelid int
typbyval boolean

3. 完成实验步骤 7.4.4，验证添加代码的执行效果，查看输出的表信息和表的属性列信息。

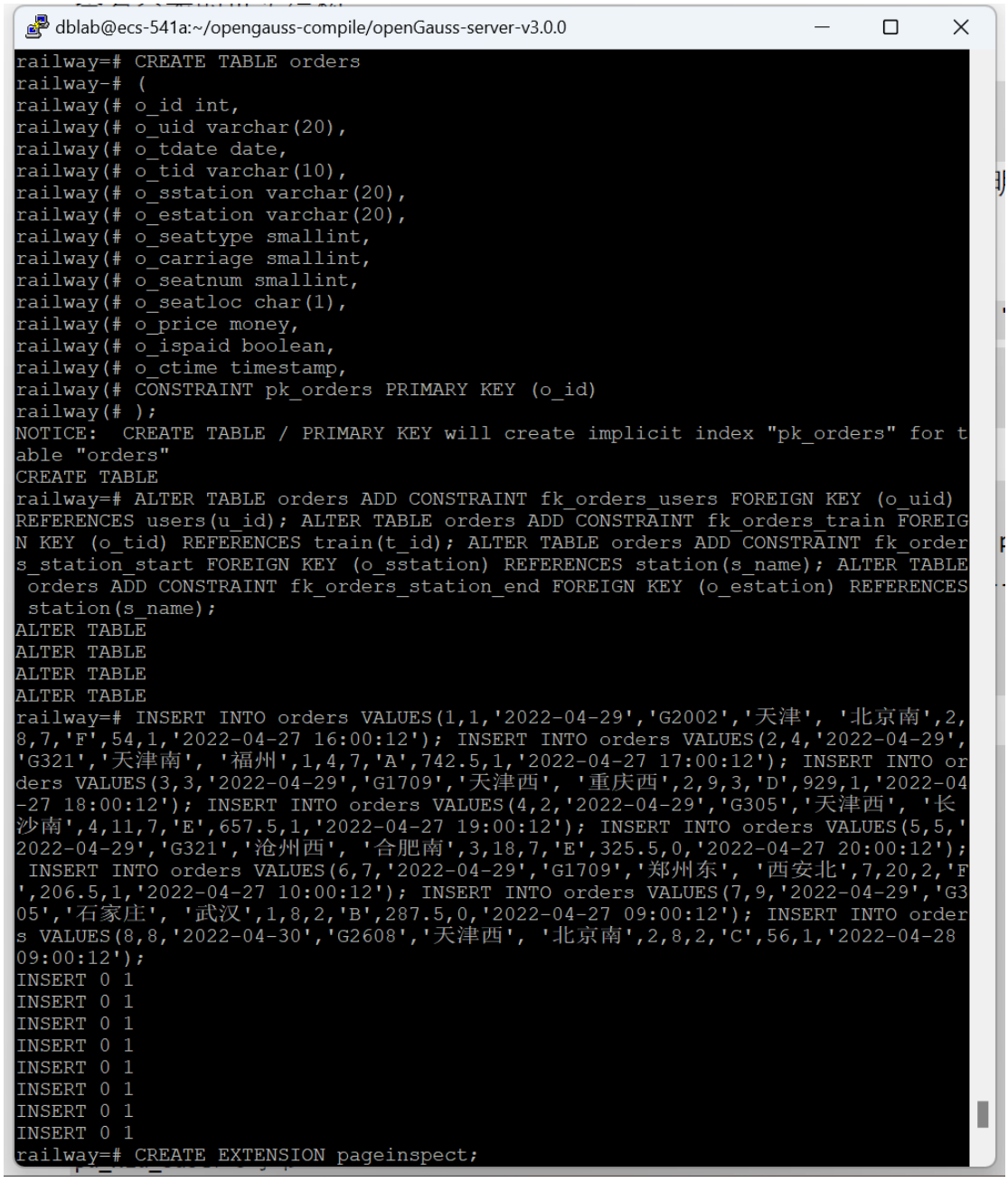
```
openGauss=# CREATE TABLE users
(
u_id varchar(20),
u_passwd varchar(20),
u_name varchar(10),
u_idnum varchar(20),
u_regtime timestamp
)
;
INFO:
relation's object id : 32772
pg_class->relname : users
pg_attribute->attname :
attr[0].attname : u_id typname : varchar typlen : -1
attr[1].attname : u_passwd typname : varchar typlen : -1
attr[2].attname : u_name typname : varchar typlen : -1
attr[3].attname : u_idnum typname : varchar typlen : -1
attr[4].attname : u_regtime typname : timestamp typlen : 8

CREATE TABLE
openGauss=#
```

# 第 8 章 表的页面存储结构

1. 完成实验步骤8.4.1节，安装pageinspect插件。

2. 完成实验步骤8.4.2节，使用gsql创建表和插入数据。




3. 完成实验步骤8.4.3节，根据分析users表页面结构的例子，分析orders表页面结构，画出orders表页面结构图（可使用Visio等画图工具）。
创建一个新的表 orders2，与 orders 表结构完全相同，只是表名不同。

执行一条插入语句，执行下列语句并查看返回结果：

```
railway=# SELECT * FROM page_header(get_raw_page('orders2', 0));
```

lsn | tli | flags | lower | upper | special | pagesize | version | prune_xid

```
-----------+-----+-------+-------+-------+---------+----------+---------+---
 0/244CE98 |   0 |     0 |    44 |  8096 |    8192 |     8192 |       6 |   16056
```

可以看到，lower 列的值恰为 44。页面大小为 8192 字节，upper 列的值为 8096，即刚插入的第 1 条元组占用了 96 字节。

```
     lsn      | tli | flags | lower | upper | special | pagesize | version | prune_xid
-----------+-----+-------+-------+-------+---------+----------+---------+---
--------
 0/2460C30 |   0 |     0 |    48 |  8000 |    8192 |     8192 |       6 |   16065
```

可以看到，lower列值此时变为了48，增加了4，验证了一个元组指针占用4字节。upper列值此时变为了8000，减少了96，验证了此表的一个元组数据占96字节。

将剩余的 7 条元组插入：

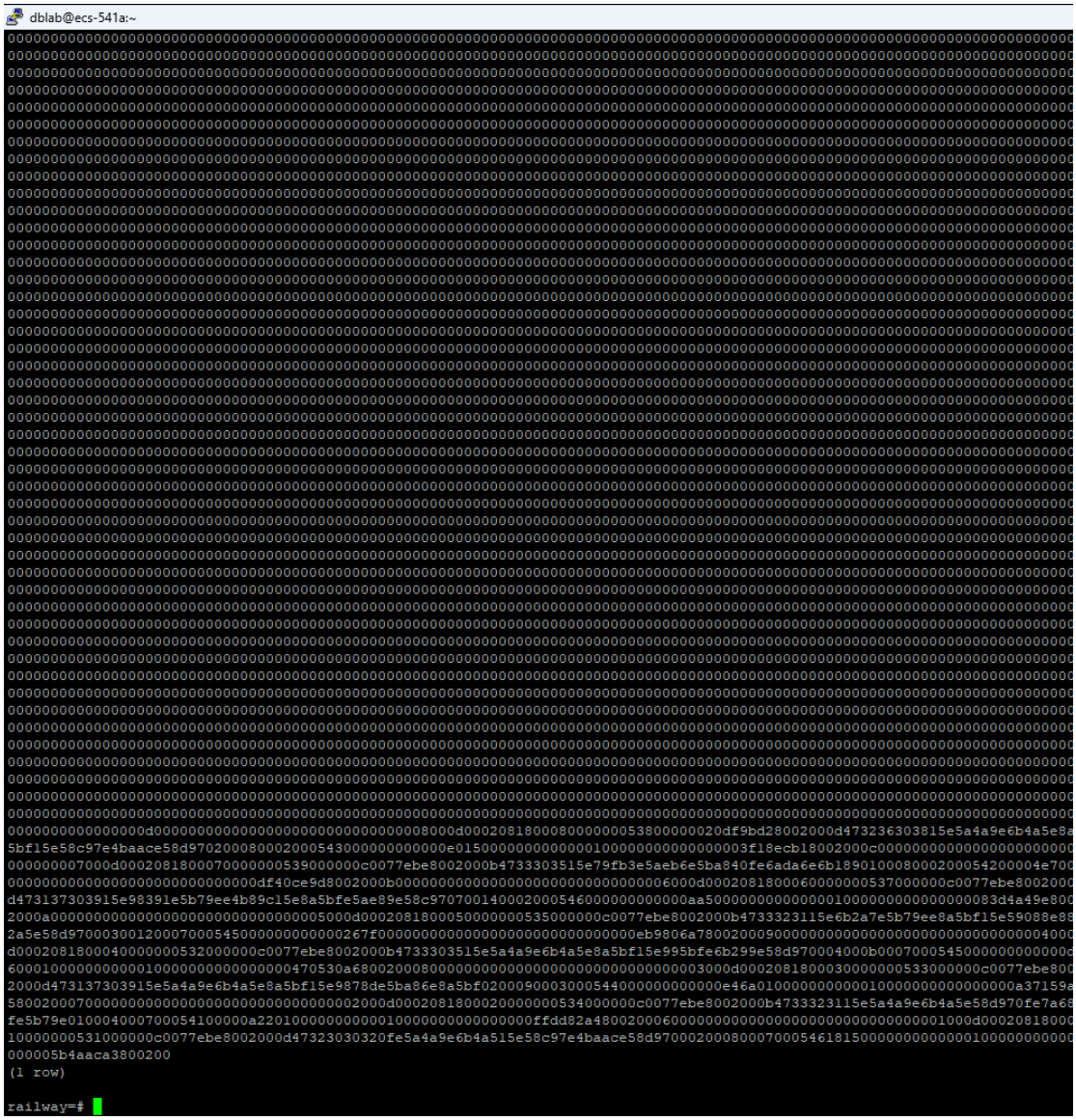
```
     lsn      | tli | flags | lower | upper | special | pagesize | version | prune_xid
-----------+-----+-------+-------+-------+---------+----------+---------+---
--------
 0/2462020 |   0 |     0 |    72 |  7384 |    8192 |     8192 |       6 |   16065
```

此时，共有 8 个元组，8 个元组指针占用 32 字节，因此 lower 列值为 72;

```
railway=# SELECT * FROM heap_page_items(get_raw_page('orders2', 0));
 lp | lp_off | lp_flags | lp_len | t_xmin | t_xmax | t_field3 | t_ctid | t_infomask2 | t_infomask | t_hoff | t_bits | t_oid
----+--------+----------+--------+--------+--------+----------+--------+-------------+------------+--------+--------+-------
  1 |   8096 |        1 |     96 |  16071 |      0 |        0 | (0,1)  |          13 |       2050 |     24 |        |
  2 |   8000 |        1 |     96 |  16072 |      0 |        0 | (0,2)  |          13 |       2050 |     24 |        |
  3 |   7896 |        1 |    104 |  16073 |      0 |        0 | (0,3)  |          13 |       2050 |     24 |        |
  4 |   7792 |        1 |    104 |  16074 |      0 |        0 | (0,4)  |          13 |       2050 |     24 |        |
  5 |   7688 |        1 |    104 |  16075 |      0 |        0 | (0,5)  |          13 |       2050 |     24 |        |
  6 |   7584 |        1 |    104 |  16076 |      0 |        0 | (0,6)  |          13 |       2050 |     24 |        |
  7 |   7488 |        1 |     96 |  16077 |      0 |        0 | (0,7)  |          13 |       2050 |     24 |        |
  8 |   7384 |        1 |    104 |  16078 |      0 |        0 | (0,8)  |          13 |       2050 |     24 |        |
(8 rows)
```

插入表中的 8 个元组数据所占字节不同，如图所示，因此 upper 列值为 8192 – 808 = 7384。

原始数据为




综合以上信息，可以得到 orders 表的页面结构图

| | | | | |
|---|---|---|---|---|
| pd_lsn= 2462020 | TLI = 0 | pd_flags = 0 | pd_lower=72 | pd_upper = 7616 |
| pd_special = 8192 | pd_pagesize = 8192<br>version = 6 | pd_prune_xid =16065 | pd_xid_base | |
| pd_multi_base | offset = 8096,len = 96 ,flag = 1 | offset = 8000,len = 96 ,flag = 1 | offset = 7896,len = 104 ,flag = 1 | offset = 7792,len = 104 ,flag = 1 |
| offset = 7688,len = 104 ,flag = 1 | offset = 7584,len = 104 ,flag = 1 | offset = 7488,len = 104 ,flag = 1 | offset = 7384,len = 104 ,flag = 1 | freespace |
| freespace | | | | |
| freespace | | | | |
| freespace | | tuple8 | tuple7 | tuple6 |
| tuple5 | tuple4 | tuple3 | tuple2 | tuple1 |

4. 使用pageinspect插件功能，对railway数据库中的其他表调用page_header和heap_page_items函数，获取对应的页头结构和元组数据描述。

TABLE station

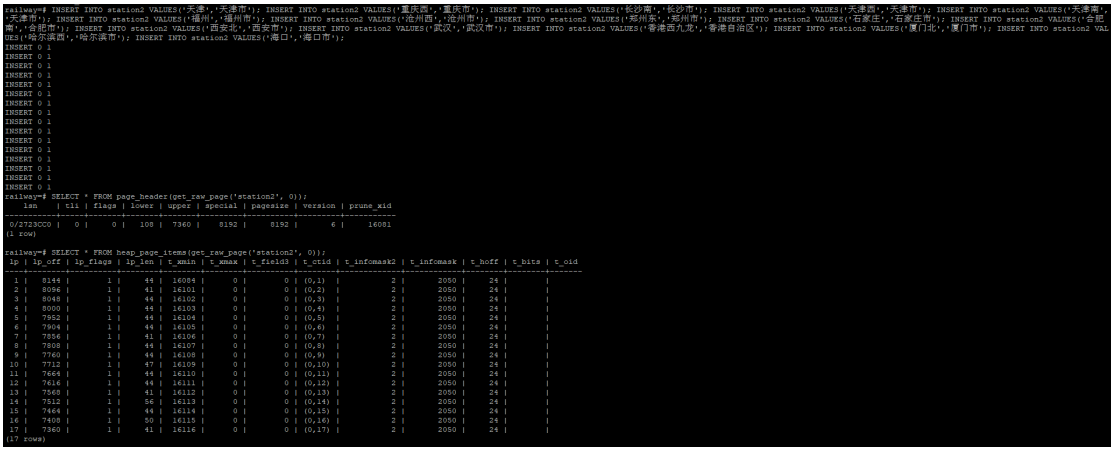



Table trainstop

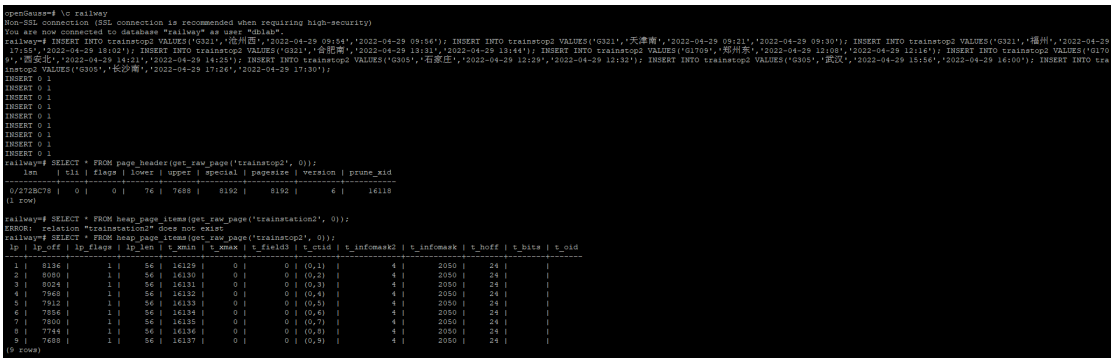



Table trainrun