

# opengauss数据库安装及数据载入

姓名	学号
崔璨明	20337025

## 1、实验内容

完成opengauss数据库的安装，并把实验用的school数据库载入opengauss数据库。

## 2、opengauss安装

首先进行opengauss的安装，我选择了在windows下通过docker来安装opengauss，然后安装opengauss的配套可视化工具data studio来进行管理，通过docker安装opengauss的过程可以参考该篇博客：

[windows下通过docker安装opengauss chainsxaw的博客-CSDN博客 windows安装opengauss](#)

主要通过`docker pull enmotech/opengauss`来拉取opengauss的镜像，然后创建opengauss的容器即可，创建完成后可以通过以下命令进行本地连接：

```
su - omm
gsql -d postgres -p 5432
```

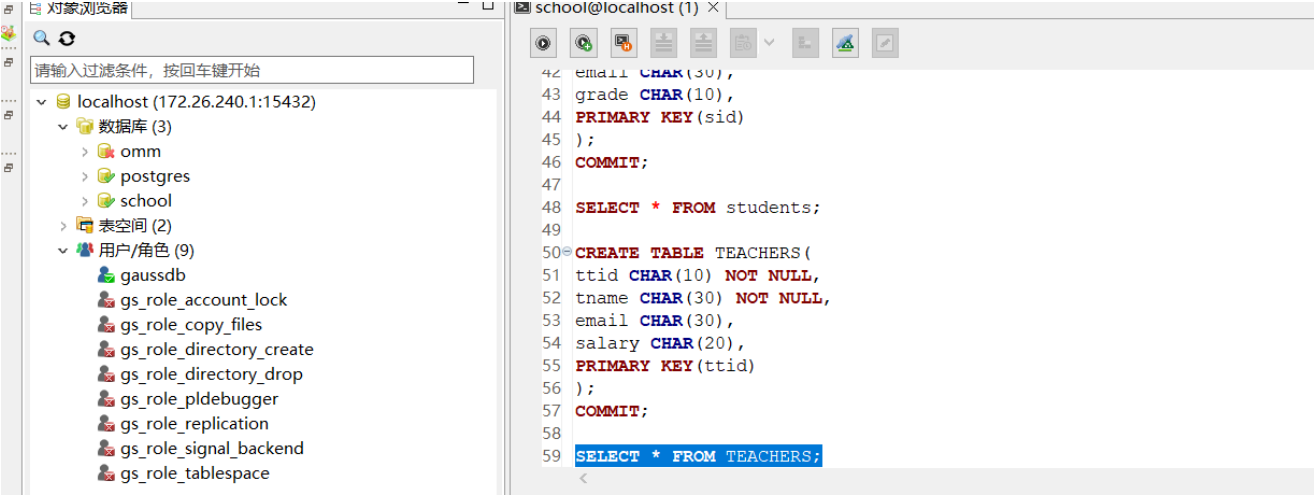
连接并进入数据库后的效果如下：

[Open in external terminal](#)

```
omm@59d3e1c0eb01:~$ gsql -d postgres -p 5432
gsql ((openGauss 3.0.0 build 02c14696) compiled at 2022-04-01 18:12:34 commit 0 last mr )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

openGauss=#
```

使用可视化管理工具data studio连接如下：



### 3、school数据库载入

由于对opengauss不是很熟悉，没有找到更高效的数据导入方法，于是我采取了以下比较繁杂的方法来进行数据导入：

- 1. 在sql sever中将school中的各个表数据导出为文本文件；
- 2. 在opengauss中创建相应的表；
- 3. 复制表中的数据，粘贴到标准输入；
- 4. 使用copy table from stdin命令将标准输入中的数据载入相应的表中。

导入结果如下：

CHOICES表：

	no	sid	ttid	cid	score
1	500000058	823069829	249596497	10037	76
2	500000253	829348273	202560416	10010	87
3	500001270	847061074	292043491	10025	92
4	500002143	860635914	238811498	10039	82
5	500002371	829785562	273189968	10028	77
6	500002715	822137137	218922066	10011	67
7	500003262	826310502	267846042	10005	90
8	500004144	817636568	253205179	10047	60
9	500005519	813520169	226385492	10023	NULL
10	500005935	801967882	234419511	10021	70
11	500005987	875434315	223646385	10048	82

COURSES表：

	cid	cname	hour
1	10001	database	96
2	10002	operating system	88
3	10003	computer graphics	48
4	10004	java	48
5	10005	c++	60
6	10006	design pattern	48
7	10007	uml	30
8	10008	data structure	60
9	10009	cryptology	36
10	10010	software engineering	50
11	10011	distributed computing	36

查询提交时间: 2022-10-15 11:50:50.815 CST 获取50行, 共19 ms [可以编辑查询结果。](#)

## STUDENTS表:

	sid	sname	email	grade
1	60000001	Lily	NULL	2001
2	60000002	Lily	NULL	1997
3	800001216	gfxrgs	hhce4@qhldj.gov	1992
4	800005753	waqcj	hlhq0h8@jdba.gov	1992
5	800006682	fiiluommh	ihzd6_k@kzvft.gov	1992
6	800006941	ogvmu	62sfbd@lrt.gov	1995
7	800007595	uxqqbkjn	cr8g@zrvgt.edu	1997
8	800009099	zapyv	jmqn8@iwaiu.org	1992
9	800009249	zyuoh	8enjrcu@upfw.org	1991
10	800010666	uwphrw	emb7k@ipp.com	1992
11	800014004	aoaahudi	ftl0oci@fits.edu	1994

查询提交时间: 2022-10-15 11:56:22.858 CST 已获取1,000行, 耗时52 ms (持续获取中...) [可以编辑查询结果。](#)

## TEACHERS表:

	ttid	tname	email	salary
1	200001000	LXL	s4zrck@pew.net	3024
2	200003125	fqmmyi	wcjc@glq.net	3928
3	200005322	udjom	gd5l8@ppmlf.gov	2500
4	200009423	jhoulvtr	12cj@pzc.gov	3286
5	200010493	xzgkflko	zcf_m@def.com	4000
6	200011899	lwujzim	s4zrck@pew.net	NULL
7	200016731	nfgrod	dprcy@qxqi.com	3819
8	200017039	qibsy	b8mtpe@enbba.edu	4179
9	200023671	zfuxiahq	lr_32t@cfo.net	3677
10	200025513	rnupx	c7ce@ayt.org	2524
11	200026895	nuyoq	xibl@izd.edu	4830

查询提交时间: 2022-10-15 11:57:01.501 CST 已获取1,000行, 耗时33 ms (持续获取中...) [可以编辑查询结果。](#)

使用的代码如下:

```
CREATE TABLE CHOICES (
  no INT NOT NULL,
  sid CHAR(10),
```

```

ttid CHAR(10),
cid CHAR(10),
score INT,
PRIMARY KEY(no)
);

CREATE TABLE COURSES(
cid CHAR(10) NOT NULL,
cname CHAR(30) NOT NULL,
hour INT ,
PRIMARY KEY(cid)
);

CREATE TABLE STUDENTS(
sid CHAR(10) NOT NULL,
sname CHAR(30) NOT NULL,
email CHAR(30),
grade CHAR(10),
PRIMARY KEY(sid)
);

CREATE TABLE TEACHERS(
ttid CHAR(10) NOT NULL,
tname CHAR(30) NOT NULL,
email CHAR(30),
salary CHAR(20),
PRIMARY KEY(ttid)
);

COMMIT;
copy choices from stdin;
copy courses from stdin;
copy students from stdin;
copy teachers from stdin;

```

可见，实现了School数据库的导入：

```

school=# \d
                                List of relations
Schema | Name      | Type  | Owner  | Storage
-----+-----+-----+-----+-----
public | choices   | table | gaussdb | {orientation=row,compression=no}
public | courses   | table | gaussdb | {orientation=row,compression=no}
public | students  | table | gaussdb | {orientation=row,compression=no}
public | teachers  | table | gaussdb | {orientation=row,compression=no}
(4 rows)

school=#

```