



数据库系统课程实验报告

实验名称:	数据库与基本表的创建、修改与删除
实验日期:	2022.10.14
实验地点:	四号楼
提交日期:	

学号:	22920202202877
姓名:	陈鑫蕾
专业年级:	数媒 2020 级
学年学期:	2022-2023 学年第一学期

1. 实验目的

理解 openGauss 中用户、数据库、模式和表等数据库对象之间的关系

掌握用户、数据库、模式和基本表的创建、修改和删除方法

掌握数据类型的选择和使用

掌握数据的导入导出方法

2. 实验内容和步骤

(1) 连接到数据库

步骤如下：

1.在数据库主节点服务器上，切换至 omm 操作系统用户环境。

```
-bash: gsqt: command not found
[root@ecs-7cda ~]# su - omm
Last login: Tue Oct 18 10:19:04 CST 2022 on pts/0
```

2.查看服务是否启动。

```
[omm@ecs-7cda ~]$ gs_om -t status
-----
cluster_name      : dbCluster
cluster_state     : Normal
redistributing    : No
-----
```

3.启动数据库服务

```
[omm@ecs-7cda ~]$ gs_om -t start
Starting cluster.
=====
[SUCCESS] ecs-7cda:
[2022-10-18 12:26:33.753][14929][][gs_ctl]: gs_ctl started,data
dir is /gaussdb/data/db1
[2022-10-18 12:26:33.756][14929][][gs_ctl]: another server mig
ht be running; Please use the restart command
=====
Successfully started.
```

4.连接数据库

```
[omm@ecs-7cda ~]$ gsql -d postgres -p 26000 -r
gsql ((openGauss 2.0.0 build 78689da9) compiled at 2021-03-31 2
1:03:52 commit 0 last mr )
Non-SSL connection (SSL connection is recommended when requirin
g high-security)
Type "help" for help.
```

(2) 创建新用户，模式以及表

1.创建新用户

```
postgres=# CREATE USER chenxinlei WITH PASSWORD "Bigdata123";
CREATE ROLE
```

2.设置权限

```
postgres=# grant all privileges to chenxinlei;
ALTER ROLE
```

3.创建名为 Sales 的数据库

```
postgres=# CREATE DATABASE Sales OWNER chenxinlei;
CREATE DATABASE
```

4.使用新用户连接到此数据库

```
[omm@ecs-7cda ~]$ gsql -d sales -p 26000 -U chenxinlei -r
Password for user chenxinlei:
```

5.创建新模式

```
sales=> CREATE SCHEMA myschema;
CREATE SCHEMA
```

6.创建表

```
sales=> CREATE TABLE employees(  
sales(> employee_id NUMBER,  
sales(> first_name VARCHAR2(255),  
sales(> last_name VARCHAR2(255),  
sales(> email VARCHAR2(255),  
sales(> phone VARCHAR2(20),  
sales(> hire_date DATE,  
sales(> manager_id NUMBER,  
sales(> job_title VARCHAR2(255)  
sales(> );  
CREATE TABLE
```

类似 11 张操作省略

(3) 为 12 张表插入示例数据。

1.将 csv 数据导入后进行查看，导入到 home/omm 下

```
[root@ecs-7cda omm]# ls  
contacts.csv      order_itmes.csv  
countries.csv     orders.csv  
customers.csv     product_categories.csv  
employees.csv     products.csv  
inventories.csv  regions.csv  
locations.csv     warehouses.csv  
[root@ecs-7cda omm]#
```

2.执行如下命令，修改文件权限

```
[root@ecs-7cda omm]# chmod 777 *.csv  
[root@ecs-7cda omm]#
```

3.传入数据，由于下面语句导入会报如下的错误，于是改用了 insert
插入数据

```
sales=> \copy customers FROM '/home/omm/customers.csv' WITH (delimiter',' ,IGNORE_EXTRA_DATA 'on');  
ERROR:  invalid input syntax for type numeric: ""CUSTOMER_ID""  
CONTEXT:  COPY customers, line 1, column customer_id: ""CUSTOMER_ID""  
sales=>
```

```

Insert into COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('AR','Argentina',2);
Insert into COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('AU','Australia',3);
Insert into COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('BE','Belgium',1);
Insert into COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('BR','Brazil',2);
Insert into COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('CA','Canada',2);
Insert into COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('CH','Switzerland',1);
Insert into COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('CN','China',3);
Insert into COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('DE','Germany',1);
Insert into COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('DK','Denmark',1);
Insert into COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('EG','Egypt',4);
Insert into COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('FR','France',1);
Insert into COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('IL','Israel',4);
Insert into COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('IN','India',3);
Insert into COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('IT','Italy',1);
Insert into COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('JP','Japan',3);
Insert into COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('KW','Kuwait',4);
Insert into COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('ML','Malaysia',3);
Insert into COUNTRIES (COUNTRY_ID,COUNTRY_NAME,REGION_ID) values ('MX','Mexico',2);

```

(4) 修改 12 张表的结构，根据上图添加相应的约束，如主码、外码。

1.为表添加主码

ALTER TABLE 表名 ADD CONSTRAINT 约束名 PRIMARY
KEY(列名);

```

sales=> ALTER TABLE orders ADD CONSTRAINT orders_PK PRIMARY KEY
(order_id);
NOTICE: ALTER TABLE / ADD PRIMARY KEY will create implicit ind
ex "orders_pk" for table "orders"
ALTER TABLE

```

2.为表添加外码

ALTER TABLE 表名 ADD CONSTRAINT 约束名 FOREIGN
KEY(列名) REFERENCES 外键表名(外键列名);

特别注意：自己给自己设定外键要先声明唯一，否则报 there is no
unique constraint matching given keys for referenced table "employees"错

```

sales=> ALTER TABLE employees ADD UNIQUE(manager_id);
NOTICE: ALTER TABLE / ADD UNIQUE will create implicit index "e
mployees_manager_id_key" for table "employees"
ALTER TABLE

```

```

sales=> ALTER TABLE employees ADD CONSTRAINT fk_employees_manag
er FOREIGN KEY(manager_id) REFERENCES employees(employee_id);
ALTER TABLE

```

类似操作省略

(5) 查询表

SELECT * FROM 表名

1.Products 表

```
45.18 | 59.99 | 5 |
94 | Western Digital WD2500AVVS | Series:AV-GP,Type:5400RPM,Capacity:250GB,Cache:8MB
12.63 | 15.55 | 5 |
99 | Seagate ST1000DX002 | Series:FireCuda,Type:Hybrid,Capacity:1TB,Cache:64MB
55.41 | 68.86 | 5 |
108 | Crucial CT275MX300SSD1 | Series:MX300,Type:SSD,Capacity:275GB,Cache:N/A
79.21 | 97.88 | 5 |
108 | Seagate ST31000340NS - FFP | Series:Barracuda ES,Type:7200RPM,Capacity:1TB,Cache:32MB
34.40 | 43.99 | 5 |
273 | Western Digital WD101KRYZ | Series:Gold,Type:7200RPM,Capacity:10TB,Cache:256MB
313.96 | 443.64 | 5 |
55 | Crucial CT525MX300SSD4 | Series:MX300,Type:SSD,Capacity:525GB,Cache:N/A
121.92 | 158.99 | 5 |
56 | Western Digital WD2500AAJS | Series:Caviar Blue,Type:7200RPM,Capacity:250GB,Cache:8MB
15.23 | 16.99 | 5 |
57 | Western Digital WD20EZRX | Series:Blue,Type:5400RPM,Capacity:2TB,Cache:64MB
58.01 | 67.34 | 5 |
95 | Western Digital WDS256G1X0C | Series:Black PCIe,Type:SSD,Capacity:256GB,Cache:N/A
85.87 | 109.99 | 5 |
96 | SanDisk SDSSDHII-480G-G25 | Series:Ultra II,Type:SSD,Capacity:480GB,Cache:N/A
102.62 | 141.56 | 5 |
97 | Kingston SV300S37A/120G | Series:SSDNow V300 Series,Type:SSD,Capacity:120GB,Cache:N/A
45.93 | 59.87 | 5 |
(288 rows)
```

2.Warehouse 表

```
sales=> SELECT * FROM WAREHOUSES;
warehouse_id | warehouse_name | location_id
-----+-----+-----
1 | Southlake, Texas | 5
2 | San Francisco | 6
3 | New Jersey | 7
4 | Seattle, Washington | 8
5 | Toronto | 9
6 | Sydney | 13
7 | Mexico City | 23
8 | Beijing | 11
9 | Bombay | 12
(9 rows)
```

3.Product_categories 表


```

sales=> SELECT * FROM PRODUCT_CATEGORIES;
category_id | category_name
-----+-----
          1 | CPU
          2 | Video Card
          3 | RAM
          4 | Mother Board
          5 | Storage
(5 rows)

```

4.Orders 表

```

76 |          17 | Shipped |          55 | 2017-02-10 00:00:00
77 |           1 | Shipped |          60 | 2017-01-02 00:00:00
78 |           2 | Pending |          64 | 2015-12-14 00:00:00
84 |           3 | Shipped |          64 | 2015-12-01 00:00:00
85 |           4 | Pending |          55 | 2016-12-01 00:00:00
86 |           5 | Pending |          60 | 2016-11-30 00:00:00
88 |           6 | Shipped |          61 | 2017-11-01 00:00:00
89 |           7 | Shipped |          55 | 2015-10-27 00:00:00
90 |           8 | Pending |          55 | 2015-10-28 00:00:00
91 |           9 | Pending |          56 | 2016-10-28 00:00:00
92 |          44 | Shipped |          59 | 2015-10-28 00:00:00
95 |          45 | Shipped |          59 | 2016-09-19 00:00:00
96 |          46 | Shipped |          56 | 2016-09-14 00:00:00
97 |          47 | Canceled |          60 | 2016-07-12 00:00:00
98 |          48 | Shipped |          55 | 2017-03-18 00:00:00
99 |          49 | Shipped |          60 | 2017-01-07 00:00:00
100 |          16 | Pending |          54 | 2017-01-05 00:00:00
103 |          17 | Pending |          64 | 2016-02-08 00:00:00
104 |          18 | Shipped |          60 | 2017-02-01 00:00:00
(105 rows)

```

5.Order_items 表

28		8		40		105.00		299.89
29		11		115		52.00		699.99
31		6		171		103.00		382.98
32		6		102		147.00		2042.69
33		8		13		75.00		704.99
34		6		101		62.00		635.99
35		5		186		41.00		1449.98
36		8		141		144.00		419.99
39		9		144		125.00		289.79
40		11		184		88.00		999.99
41		9		55		61.00		150.99
42		7		139		149.00		65.92
46		11		89		119.00		749.99
51		2		7		64.00		680.99
53		4		177		96.00		1759.99
58		12		62		148.00		789.79
59		9		154		82.00		1204.98
64		2		126		52.00		640.99
65		6		262		118.00		1449.99
66		9		279		90.00		1318.99
67		10		239		45.00		739.99
68		9		177		84.00		1759.99
69		9		261		55.00		1504.99

(665 rows)

6.Location 表

sales=> SELECT * FROM LOCATIONS;						
location_id		address	postal_code	city	state	country_id
1		1297 Via Cola di Rie	00989	Roma		IT
2		93091 Calle della Testa	10934	Venice		IT
3		2017 Shinjuku-ku	1689	Tokyo	Tokyo Prefecture	JP
4		9450 Kamiya-cho	6823	Hiroshima		JP
5		2014 Jabberwocky Rd	26192	Southlake	Texas	US
6		2011 Interiors Blvd	99236	South San Francisco	California	US
7		2007 Zagora St	50098	South Brunswick	New Jersey	US
8		2004 Charade Rd	98199	Seattle	Washington	US
9		147 Spadina Ave	M5V 2L7	Toronto	Ontario	CA
10		6092 Boxwood St	Y9W 9T2	Whitehorse	Yukon	CA
11		40-5-12 Laogianggen	198518	Beijing		CN
12		1298 Vileparle (E)	498231	Bombay	Maharashtra	IN
13		12-98 Victoria Street	2981	Sydney	New South Wales	AU
14		198 Clementi North	540198	Singapore		SG
15		8204 Arthur St		London		UK
16		Magdalen Centre, The Oxford Science Park	OX9 9ZB	Oxford	Oxford	UK
17		9782 Chester Road	09629850293	Stretford	Manchester	UK
18		Schwanthalerstr. 7031	80925	Munich	Bavaria	DE
19		Rua Frei Caneca 1360	01307-002	Sao Paulo	Sao Paulo	BR
20		20 Rue des Corps-Saints	1730	Geneva	Geneve	CH
21		Murttenstrasse 921	3095	Bern	BE	CH
22		Pieter Breughelstraat 837	3029SK	Utrecht	Utrecht	NL

7.Inventories 表

175		8		86
182		8		128
184		8		128
185		8		129
188		8		137
189		8		137
190		8		137
191		8		138
193		8		138
195		8		139
196		8		139
198		8		140
199		8		141
200		8		141
201		8		142
203		8		142
204		8		142
205		8		142
207		8		121
(1112 rows)				

8.employees 表

Clerk	34		Seth		Foster		seth.foster@example.com		650.127.1934		2016-06-14 00:00:00		23		Stock
Clerk	35		Carter		Gonzales		carter.gonzales@example.com		650.127.1834		2016-08-26 00:00:00		23		Stock
Clerk	36		Felix		Bryant		felix.bryant@example.com		650.127.1734		2016-12-12 00:00:00		23		Stock
Clerk	37		Ibrahim		Alexander		ibrahim.alexander@example.com		650.127.1634		2016-02-06 00:00:00		23		Stock
Clerk	38		Sonny		Russell		sonny.russell@example.com		650.121.1234		2016-07-14 00:00:00		24		Stock
Clerk	39		Kian		Griffin		kian.griffin@example.com		650.121.2034		2016-10-26 00:00:00		24		Stock
Clerk	40		Caleb		Diaz		caleb.diaz@example.com		650.121.2019		2016-02-12 00:00:00		24		Stock
Clerk	25		Ronnie		Perry		ronnie.perry@example.com		650.123.5234		2016-11-16 00:00:00		1		Stock
Manager	24		Callum		Jenkins		callum.jenkins@example.com		650.123.4234		2016-10-10 00:00:00		1		Stock
Manager	23		Jackson		Coleman		jackson.coleman@example.com		650.123.3234		2016-05-01 00:00:00		1		Stock
Manager	22		Liam		Henderson		liam.henderson@example.com		650.123.2234		2016-04-10 00:00:00		1		Stock
Manager	21		Jaxon		Ross		jaxon.ross@example.com		650.123.1234		2016-07-18 00:00:00		1		Stock
(107 rows)															

9.Customers 表

	2400.00			
143	Intel	9435 Washington Blvd N # M, Laurel, MD	http://www.intel.com	
	2400.00			
147	Pfizer	206 S Broadway # 707, Rochester, MN	http://www.pfizer.com	
	2400.00			
150	FedEx	2215 Oak Industrial Dr Ne, Grand Rapids, MI	http://www.fedex.com	
	2400.00			
153	New York Life Insurance	215 4Th Ave Se, Cedar Rapids, IA	http://www.newyorklife.com	
	2500.00			
157	Ingram Micro	4811 S 76Th St # 205, Milwaukee, WI	http://www.ingrammicro.com	
	3600.00			
161	American Airlines Group	800 Carter St, Rochester, NY	http://www.aa.com	
	3600.00			
165	Johnson Controls	6Th And Master St, Philadelphia, PA	http://www.johnsoncontrols.com	
	3600.00			
170	Goldman Sachs Group	23985 Bedford Rd N, Battle Creek, MI	http://www.gs.com	
	3700.00			
174	Oracle	701 Seneca St, Buffalo, NY	http://www.oracle.com	
	5000.00			

(319 rows)

10.Countries 表

AR	Argentina	2
AU	Australia	3
BE	Belgium	1
BR	Brazil	2
CA	Canada	2
CH	Switzerland	1
CN	China	3
DE	Germany	1
DK	Denmark	1
EG	Egypt	4
FR	France	1
IL	Israel	4
IN	India	3
IT	Italy	1
JP	Japan	3
KW	Kuwait	4
ML	Malaysia	3
MX	Mexico	2
NG	Nigeria	4
NL	Netherlands	1
SG	Singapore	3
UK	United Kingdom	1
US	United States of America	2
ZM	Zambia	4
ZW	Zimbabwe	4

(25 rows)

11.Contacts 表

297	Murphy	Robertson	murphyrobertson@kctenergy.com	+1 80 012 3689	297
298	Elly	Salinas	elly.salinas@kiewit.com	+1 80 012 3687	298
299	Ginny	Carlson	ginny.carlson@orate.murphyusa.com	+1 80 012 3689	299
300	Mozelle	Schneider	mozelle.schneider@cbre.com	+1 80 012 3695	300
301	Adam	Jacobs	adam.jacobs@univar.com	+1 80 012 3699	301
302	Vernia	Hayes	verniahayes@drhorton.com	+1 80 012 3701	302
303	Ismael	Solomon	ismael.solomon@elcompanies.com	+1 80 012 3703	303
304	Ivey	Rutledge	ivey.rutledge@praxair.com	+1 80 012 3709	304
305	Jame	Terrell	jame.terrell@biogen.com	+1 80 012 3711	305
306	Daina	Combs	daina.combs@mosaicco.com	+1 80 012 3715	306
307	Lashunda	Davidson	lashunda.davidson@statestreet.com	+1 80 012 3717	307
308	Arlette	Thornton	arlette.thornton@unum.com	+1 80 012 3719	308
309	Darron	Robertson	darron.robertson@reynoldsamerican.com	+1 80 012 3723	309
310	Maple	Barnett	maple.barnett@core-mark.com	+1 80 012 3725	310
311	Charlesie	Carey	charlesie.carey@grouplauto.com	+1 80 012 3731	311
312	Frank	Shannon	frank.shannon@henryschein.com	+1 80 012 3733	312
313	Josiah	Beasley	josiah.beasley@hertz.com	+1 80 012 3735	313
314	Annabelle	Butler	annabelle.butler@thrivent.com	+1 80 012 3737	314
315	Ed	Mueller	ed.mueller@nscorp.com	+1 80 012 3739	315
316	Colleen	Estrada	colleen.estrada@rgare.com	+1 80 012 3741	316
317	Nikia	Kent	nikia.kent@psag.com	+1 80 012 3745	317
318	Amber	Brady	amber.brady@bc-a-m.com	+1 80 012 3837	318
319	Wendell	Massey	wendell.massey@bbt.com	+1 80 012 3839	319

(319 rows)

12.Regions 表

```
sales=> SELECT * FROM REGIONS;
 region_id |      region_name
-----+-----
          1 | Europe
          2 | Americas
          3 | Asia
          4 | Middle East and Africa
(4 rows)
```

(6) 删除表，模式，数据库，用户

1.删除表

```
sales=> drop table regions cascade;
DROP TABLE
```

2.删除模式

```
sales=> drop schema myschema cascade;
```

```
DROP SCHEMA
```

3.删除数据库

```
postgres=# drop database sales;
DROP DATABASE
```

4.删除用户

```
postgres=# drop user chenxinlei;
DROP ROLE
```

3. 实验总结

3.1 完成的工作

- 1) 理解 Sales 数据库中各表及各表之间的联系
- 2) 创建新用户、数据库和模式；--**新用户名建议用自己名字的拼音**
- 3) 分别创建 12 张表，创建时不添加约束。
- 4) 修改 12 张表的结构，根据上图添加相应的约束，如主码、外码。
- 5) 为 12 张表插入示例数据。
- 6) 分别查询 12 张表的数据。
- 7) 完成上述任务后删除用户、数据库、模式和所有的表。

3.2 对实验的认识

掌握了数据库的基本用法；基本的 sql 语句；数据如何导入数据库

3.3 遇到的困难及解决方法

在导入数据时，会存在如下报错

```
sales=> \copy customers FROM '/home/omm/customers.csv' WITH (delimiter',' ,IGNORE_EXTRA_DATA 'on');
ERROR:  invalid input syntax for type numeric: ""CUSTOMER_ID""
CONTEXT:  COPY customers, line 1, column customer_id: ""CUSTOMER_ID""
sales=> []
```

所以改用了 sql 语句 Insert 导入数据，后续发现可以通过修改表属性的数据类型，将所有的 number 类型修改成 varchar2 () 类型，即可正常导入。