浙江工业大学

数据库原理及应用实验报告

(2021 级)

实验题目　实验 1 数据定义

学生姓名　温家伟

学生学号　202103151422

学科(专业)　大数据分析 2101 班

所在学院　理学院

提交日期　2023 年 3 月 9 日

实验 1、数据定义

1.1 实验目的

熟悉SQL的数据定义语言，能够理解并熟练地使用SQL语句来创建、修改和删除数据库、模式和基本表，创建和取消索引，特别是各种参数的具体含义和使用方法。掌握SQL 语句常见语法错误的调试方法。

1.2 实验内容

(1) 使用CREATE语句创建数据库XXXDB，该数据库为xxxuser所属。

(2) 使用SQL语句创建SCHEMA并授权给用户xxxuser。

(3) 使用CREATE语句创建基本表。

(4) 更改基本表的定义: 增加列，删除列，修改列的数据类型。

(5) 创建表的升、降序索引。

(6) 删除基本表的约束、基本表的索引或基本表。

其中xxx是自己姓名拼音缩写字母，如黄德才，则为hdc。

1.3 实验步骤

(0)启动openGauss数据库管理系统:

a) 首先以管理员身份启动Vbox;

b) 启动openEuler操作系统虚拟机镜像系统;

c) 通过putty远程登录到虚拟机系统，其中主机IP一般是192.168.56.129，可以通过ifconfig查看。

```
// 此处我是用xshell登录的
```

d) 以事先已经创建好的数据库系统用户omm身份启动openGauss机群。

切换到登录用户omm

su – omm # root切换其他用户不用输密码，普通用户直接切换密码需要知道那个用户的登录密码

//启动服务命令

gs_om -t start

```
[omm@db1 ~]$ gs_om -t start
Starting cluster.
=========================================
=========================================
Successfully started.
```

e) 连接数据库

gsql -d postgres -p 26000 –r

```
[omm@db1 ~]$ gsql -d postgres -p 26000 -r
gsql ((openGauss 1.0.1 build a362883b) compiled at 2020-10-14 02:00:13 commit 0
last mr  )
```

注意：可能会提醒omm密码过期

NOTICE : The password has been expired, please change the password.

```
NOTICE : The password has been expired, please change the password.
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.
```

连接数据库时，omm用户密码为：openGauss@123，可以先修改密码，比如新密码修改为openGausszjut@123（建议用户自定义密码）。可以执行

alter role omm identified by 'openGausszjut@123' replace 'openGauss@123';

```
postgres=# ALTER ROLE omm IDENTIFIED BY 'ahwei@520' REPLACE 'openGauss@123';
ALTER ROLE
```

f) 创建数据库用户并授权

创建数据库用户：

CREATE USER ylhuser WITH PASSWORD "openGauss@123";

```
postgres=# create user ahwei with password "ahwei@520";
CREATE ROLE
```

授权ylhuser用户为系统管理员用户：

alter user ylhuser sysadmin;

```
postgres=# alter user ahwei sysadmin;
ALTER ROLE
```

(1)使用CREATE语句创建数据库XXXDB:

- 若数据库XXXDB存在的话将删除

```
postgres=# drop database if exists ahweiDB;
NOTICE:  database "ahweidb" does not exist, skipping
DROP DATABASE
```

- 采用字符集UTF-8创建名为XXXDB 的数据库，并设置xxxuser为其属主

```
postgres=# create database ahweiDB encoding='utf-8' owner ahwei;
CREATE DATABASE
```

- 切换到创建好的数据库中执行命令：

```
postgres=# \c ahweiDB;
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "ahweidb" as user "omm".
```

- 若创建时没有设置Owner，可以更改

```
ahweidb=# alter database ahweiDB owner to omm;
ALTER DATABASE
```

(2)使用SQL语句查看当前模式搜索路径:

创建SCHEMA并授权给用户xxxuser所有，设置当前会话的搜索路径为xxxuser 模式、public 模式，随后创建的基本表就会自动创建xxxuser 模式下。

- 查看当前模式搜索路径:

```
ahweidb=# show search_path;
  search_path
----------------
 "$user",public
(1 row)
```

- 若数据库XXXDB中存在模式xxxSchema的话将删除

```
ahweidb=# drop schema if exists ahweischema;
NOTICE:  schema "ahweischema" does not exist, skipping
DROP SCHEMA
```

- 在数据库XXXDB中创建名为xxxSchema的模式，并授权给用户xxxuser所有。

```
ahweidb=# create schema ahweischema authorization ahwei;
CREATE SCHEMA
```

- 设置当前会话的搜索路径为xxxSchema模式、Public 模式，随后创建的基本表就会自动创建xxxuser 模式下。

```
ahweidb=# SET SEARCH_PATH TO ahweiSchema, Public;
SET
```

- 再次查看当前模式搜索路径

```
ahweidb=# SHOW SEARCH_PATH;
     search_path
---------------------
 ahweischema, public
(1 row)
```

(3)用SQL语句创建关系数据库基本表:

学生表Students(Sno,Sname, Semail,Scredit,Sroom);

教师表Teachers(Tno,Tname,Temail,Tsalary);

课程表Courses(Cno,Cname,Ccredit);

成绩表STC(Sno,Tno,Cno, Score);

其中：Sno、Tno、Cno分别是表Students、表Teachers、表Courses的主键，具有唯一性约束，Scredit具有约束"大于等于0"； STC中的Sno,Tno,Cno是外键，它们共同组成STC的主键。

在XXXDB 数据库的xxxSchema模式中创建4 个基本表。

```
ahweidb=# DROP TABLE IF EXISTS STC;
NOTICE:  table "stc" does not exist, skipping
DROP TABLE

ahweidb=# DROP TABLE IF EXISTS Students;
NOTICE:  table "students" does not exist, skipping
DROP TABLE

ahweidb=# CREATE TABLE Students
ahweidb-# (Sno VARCHAR(6),
ahweidb(#  Sname VARCHAR(20) NOT NULL,
ahweidb(#  Semail VARCHAR(50),
ahweidb(#  Scredit DECIMAL( 5,1 ),
ahweidb(#  Sroom VARCHAR(5),
ahweidb(#    CONSTRAINT PK_Stu PRIMARY KEY(Sno),
ahweidb(#    CONSTRAINT CK_Student_Scredit CHECK(Scredit>=0)
ahweidb(#    );
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "pk_stu" for
table "students"
CREATE TABLE

ahweidb=# DROP TABLE IF EXISTS Teachers;
NOTICE:  table "teachers" does not exist, skipping
DROP TABLE

ahweidb=# CREATE TABLE Teachers
ahweidb-# (Tno VARCHAR(6),
ahweidb(#  Tname VARCHAR(20) NOT NULL,
ahweidb(#  Temail VARCHAR(50),
ahweidb(#  Tsalary DECIMAL( 5,1 ),
ahweidb(#    CONSTRAINT PK_Tea PRIMARY KEY(Tno)
ahweidb(#    );
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "pk_tea" for
table "teachers"
CREATE TABLE

ahweidb=# DROP TABLE IF EXISTS Courses;
NOTICE:  table "courses" does not exist, skipping
DROP TABLE

ahweidb=# CREATE TABLE Courses/*列级完整性约束条件*/
ahweidb-# (Cno VARCHAR(6),
ahweidb-# (Cno VARCHAR(6),/*Cno为主键*/
ahweidb(#  Cname VARCHAR(20) NOT NULL,
ahweidb(#  Cname VARCHAR(20) NOT NULL,/*Cname不能为空值*/
ahweidb(# Ccredit DECIMAL( 5,1 ),
ahweidb(#    CONSTRAINT PK_Cou PRIMARY KEY(Cno)
ahweidb(#   );
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "pk_cou" for
table "courses"
CREATE TABLE

ahweidb=# DROP TABLE IF EXISTS STC;
NOTICE:  table "stc" does not exist, skipping
DROP TABLE
```

```
ahweidb=# CREATE TABLE STC/*列级完整性约束条件*/
ahweidb-# (Sno VARCHAR(6) ,
ahweidb(#  Tno VARCHAR(6) ,
ahweidb(#  Cno VARCHAR(6) ,
ahweidb(#  Cno VARCHAR(6) ,
ahweidb(#  Score DECIMAL( 5,1 ),
ahweidb(#  CONSTRAINT PK_Rep PRIMARY KEY(Sno, Tno, Cno),/*Sno,Tno,Cno为主键*/
ahweidb(#  CONSTRAINT FK_Stu_Rep FOREIGN KEY(Sno) REFERENCES Students,
ahweidb(#  CONSTRAINT FK_Tea_Rep FOREIGN KEY(Tno) REFERENCES Teachers,
ahweidb(#    CONSTRAINT FK_Cou_Rep FOREIGN KEY(Cno) REFERENCES Courses
ahweidb(# );
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "pk_rep" for
table "stc"
CREATE TABLE
```

(4) 更改表Students：

增加属性Ssex(类型是VARCHAR，长度为3)，取消Scredit“大于等于0”约束。分别把表Students中的属性Sname、Teachers中的属性Tname、Courses中的属性Cname的数据类型改成长度为30。

- 更改表Students：增加属性Ssex(类型是VARCHAR，长度为3)

```
ahweidb=# ALTER TABLE IF EXISTS  Students  ADD COLUMN Ssex  VARCHAR(3);
ALTER TABLE
```

- 取消Scredit“大于等于0”约束

```
ahweidb=# ALTER TABLE IF EXISTS Students DROP CONSTRAINT
CK_Student_Scredit;
ALTER TABLE
```

- 加上

```
ALTER TABLE IF EXISTS Students  ADD CONSTRAINT CK_Student_Scredit
CHECK(Scredit>=0);
ALTER TABLE
```

- 修改表Students中的属性Sname的数据类型改成长度为30。

```
ALTER TABLE IF EXISTS  Students  MODIFY  Sname  VARCHAR(30);
ALTER TABLE
```

- 修改表Teachers中的属性Tname的数据类型改成长度为30。

```
ahweidb=# ALTER TABLE IF EXISTS  Teachers  MODIFY  Tname  VARCHAR(30);
ALTER TABLE
```

- 修改表Courses中的属性Cname的数据类型改成长度为30。

```
ahweidb=# ALTER TABLE IF EXISTS  Courses  MODIFY  Cname  VARCHAR(30);
ALTER TABLE
```

(5)删除表Students的一个属性Sroom:

删除前:

```
ahweidb=# \d students;
           Table "public.students"
 Column  |         Type          | Modifiers
---------+-----------------------+-----------
 sno     | character varying(6)  | not null
 sname   | character varying(30) | not null
 semail  | character varying(50) |
 scredit | numeric(5,1)          |
 sroom   | character varying(5)  |
 ssex    | character varying(3)  |
Indexes:
    "pk_stu" PRIMARY KEY, btree (sno) TABLESPACE pg_default
Check constraints:
    "ck_student_scredit" CHECK (scredit >= 0::numeric)
Referenced by:
    TABLE "stc" CONSTRAINT "fk_stu_rep" FOREIGN KEY (sno) REFERENCES
students(sno)
```

删除:

```
ahweidb=# alter table students drop column sroom;
ALTER TABLE
```

删除后:

```
ahweidb=# \d students;
           Table "public.students"
 Column  |         Type          | Modifiers
---------+-----------------------+-----------
 sno     | character varying(6)  | not null
 sname   | character varying(30) | not null
 semail  | character varying(50) |
 scredit | numeric(5,1)          |
 ssex    | character varying(3)  |
Indexes:
    "pk_stu" PRIMARY KEY, btree (sno) TABLESPACE pg_default
Check constraints:
    "ck_student_scredit" CHECK (scredit >= 0::numeric)
Referenced by:
    TABLE "stc" CONSTRAINT "fk_stu_rep" FOREIGN KEY (sno) REFERENCES
students(sno)
```

(6)删除表STC，然后重建表STC:

删除表STC:

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

```
ahweidb=# drop table stc;
DROP TABLE
```

```
ahweidb=# \d
                        List of relations
 Schema |   Name   | Type  | Owner |             Storage
--------+----------+-------+-------+---------------------------------
 public | courses  | table | omm   | {orientation=row,compression=no}
 public | students | table | omm   | {orientation=row,compression=no}
 public | teachers | table | omm   | {orientation=row,compression=no}
(3 rows)
```

重建表STC：

```
ahweidb=# CREATE TABLE property(
ahweidb(#         pro_id INT PRIMARY KEY,
ahweidb(#         pro_c_id INT NOT NULL,
ahweidb(#         pro_pif_id INT NOT NULL,
ahweidb(#         pro_type INT NOT NULL,
ahweidb(#         pro_status CHAR(20),
ahweidb(#         pro_quantity INT,
ahweidb(#         pro_income INT,
ahweidb(#         pro_purchase_time DATE
ahweidb(# );
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "property_pkey"
for table "property"
CREATE TABLE
```

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

步骤 1 创建索引。

在普通表property上创建索引。

```
ahweidb=# CREATE INDEX idx_property ON property(pro_c_id DESC, pro_income,
pro_purchase_time);
CREATE INDEX
```

步骤 2 重命名索引。

在普通表property上重建及重命名索引。

重建索引。

```
ahweidb=# DROP INDEX idx_property;
DROP INDEX
ahweidb=# CREATE INDEX idx_property ON property(pro_c_id DESC, pro_income,
pro_purchase_time);
CREATE INDEX
```

重命名索引。

```
ahweidb=# ALTER INDEX idx_property RENAME TO idx_property_temp;
ALTER INDEX
```

步骤 3 删除索引。

删除索引idx_property_temp。

```
ahweidb=# DROP INDEX idx_property_temp;
DROP INDEX
```

```
ahweidb=# \d property;
                    Table "public.property"
      Column       |              Type               | Modifiers
-------------------+---------------------------------+-----------
 pro_id            | integer                         | not null
 pro_c_id          | integer                         | not null
 pro_pif_id        | integer                         | not null
 pro_type          | integer                         | not null
 pro_status        | character(20)                   |
 pro_quantity      | integer                         |
 pro_income        | integer                         |
 pro_purchase_time | timestamp(0) without time zone  |
Indexes:
    "property_pkey" PRIMARY KEY, btree (pro_id) TABLESPACE pg_default
```

(7)为Courses表创建按Cno降序排列的索引：

```
ahweidb=# create index cno_index on courses(cno desc);
CREATE INDEX
```

(8) 为Students表创建按Sno升序排列的索引:

```
ahweidb=# create index sno_index on students(sno desc);
CREATE INDEX
```

(9)创建表Students的按Sname升序排列的唯一性索引:

```
ahweidb=# create unique index sname_index on students(sname desc);
CREATE INDEX
```

(10)删除Students表Sno的升序索引，删除Courses表Cno的降序索引:

```
ahweidb=# drop index cno_index;
DROP INDEX
ahweidb=# drop index sno_index;
DROP INDEX
```

注意正常关闭数据库机群:

```
ahweidb=# \q
```

```
[omm@db1 ~]$ gs_om -t stop
Stopping cluster.
=========================================
Successfully stopped cluster.
=========================================
End stop cluster.
```