|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 《数据库系统概论》实验报告 | | | | | |
| 题目：SQL语言的DDL | 姓名 | 刘爱兵 | 日期 | 2020.10.22 | |
| 实验内容与完成情况：  **一、实验内容：**  1、启动数据库服务软件SQL Server 2008的查询分析器，用如下语句对表进行操作，详细的语法格式参看课本相应章节：  Create Table 建表  Drop Table 删除表  Alter Table 更改表  如下语句对索引进行操作，详细的语法格式参看课本相应章节：  Create Index 建立索引  Drop Index 删除索引  2、具体要求、步骤如下：  使用SQLServer查询分析器以语句方式创建数据库的方法所如下：   1. 在如图所示的界面中，单击工具栏的“新建查询（N）”，启动查询分析器（如图示）。   （2）在如图所示的界面中，利用查询分析器，可以直接使用语句创建数据库、数据表和视图等。其具体方法如下：  ◆在查询分析器中，直接输入SQL语句；  ◆单击语句语法分析按钮“√”，检查分析SQL语句是否正确；  ◆单击执行按钮“！执行（X）”，运行查询分析器中的语句序列。  创建数据库的语句格式：  CREATE DATABASE <数据库名>  [ON  ( NAME = <数据表名>,  FILENAME = <盘符\路径\数据库存储名称>)  [LOG ON  ( NAME = <日志文件名>,  FILENAME = <盘符\路径\日志文件存储名称>)]]  创建数据库学生选课的语句：  CREATE DATABASE 学生选课  ON  ( NAME = '学生选课\_dat',  FILENAME = 'D:\data\学生选课\_dat.mdf')  LOG ON  ( NAME = '学生选课\_Log',  FILENAME = 'D:\data\学生选课\_Log.ldf ')      或者  CREATE DATABASE 学生选课  ON  ( NAME = '学生选课\_dat',  FILENAME = 'D:\data\学生选课\_dat.mdf')       或者  CREATE DATABASE 学生选课  在默认存储设备的默认路径上，创建默认的逻辑数据库名称（学生选课.MDF）和逻辑日志名称（学生选课\_LOG.LDF）的数据库学生选课。其默认路径如下：  C:\Program Files\Microsoft SQL Server\MSSQL\Data\学生选课.mdf  C:\Program Files\Microsoft SQL Server\MSSQL\Data\学生选课\_log.ldf  注意：上述三个格式中，第一种格式，可以按照指定的逻辑数据库名称和逻辑日志名称，在指定存储设备的指定路径上创建数据库；第二种格式，可以按照指定的逻辑数据库名称和默认的逻辑日志名称（数据库名称\_LOG.LDF），在指定存储设备的指定路径上创建数据库；第三种格式，可以按照默认的逻辑数据库名称（数据库名称.MDF）和逻辑日志名称（数据库名称\_LOG.LDF），在默认存储设备的默认路径（C:\Program Files\Microsoft SQL Server\MSSQL\Data\）上创建数据库。  删除数据库的语句格式：  DROP DATABASE <数据库名>[，<数据库名>，…]  输出对象的Id和名称可以使用如下SELECT语句：  SELECT <表达式>[，…，<表达式>]  创建数据表的语句格式：  CREATE TABLE <数据表名>  (<列名> <数据类型>[ <列级完整性约束条件> ]  [，<列名> <数据类型>[ <列级完整性约束条件>] ] …  [，<表级完整性约束条件> ])  说明：  <数据表名>：所要定义的数据表的名称；  <列名>：组成数据表的各个数据项的名称；  <列级完整性约束条件>：数据项的完整性约束条件；  <表级完整性约束条件>：数据表的完整性约束条件；  <数据类型>：数据项的数据类型。常用的数据类型如下表所示。  表1  SQL Server 2008常用数据类型  BIT, SMALLINT, INT, BIGINT 整型  MONEY, SMALLMONEY 货币型  FLOAT,REAL 浮点型  DATETIME, SMALLDATETIME 日期时间型  IMAGE 二进制图像型  CHAR（n） 长度为n的字符型  创建数据库学生选课的相应数据表Student、Course和SC的语句如下：  CREATE TABLE Student(  SNo   CHAR(10) PRIMARY KEY NOT NULL,  SName CHAR(20) NOT NULL,  SSex  CHAR(2),  SAge  INT)             或者  CREATE TABLE Student(  SNo   CHAR(10) PRIMARY KEY NOT NULL,  SName CHAR(20) NOT NULL,  SSex  CHAR(2),  SAge  INT CHECK (SAge>1 and SAge<100))        或者  CREATE TABLE Student(  SNo   CHAR(10) PRIMARY KEY NOT NULL,  SName CHAR(20) NOT NULL,  SSex  CHAR(2),  SAge  INT,  CONSTRAINT SAge1\_100 CHECK (SAge>1 and SAge<100))      或者  CREATE TABLE Student(  SNo   CHAR(10) NOT NULL,  SName CHAR(20) CONSTRAINT SName\_NotNull NOT NULL,  SSex  CHAR(2),  SAge  INT,  CONSTRAINT SNo\_PK PRIMARY KEY(SNo),  CONSTRAINT SName\_Unique UNIQUE(sname),  CONSTRAINT SAge1\_100 CHECK (sage>1 and sage<100))      或者  CREATE TABLE Course(  CNo   CHAR(10) PRIMARY KEY NOT NULL,  CName CHAR(20) NOT NULL,  CCredit INT)                    或者  CREATE TABLE SC (  SNo CHAR(10) NOT NULL,  CNo CHAR(10) NOT NULL,  Grade FLOAT,  PRIMARY KEY (SNo, CNo),  FOREIGN KEY (SNo) REFERENCES Student(SNo),  FOREIGN KEY (CNo) REFERENCES Course(CNo))  注意：在创建数据表之前，先使用USE语句打开相应的数据库。或者在SQL Server的管理工作室的工具栏的“可用数据库”下拉列表框中，选择相应的数据库。  技巧：在创建数据表之前，如果该数据表已经存在，可以先使用OBJECT\_ID检测其存在性，若非空，则可以先删除该数据表，然后再建立。具体实现方法如下：  ALTER TABLE <表名>  ADD <新列名> <数据类型> [ 完整性约束 ]  修改数据表的数据项的语句格式：  ALTER TABLE <表名>  ALTER COLUMN <列名> <数据类型>  删除数据表的数据项的语句格式：  ALTER TABLE <表名>  DROP COLUMN <列名> | CONSTRAINT <约束名称>  删除数据表的数据项约束的语句格式：  ALTER TABLE <表名>  DROP CONSTRAINT <约束名称>  删除数据表的语句格式：  DROP TABLE <数据表名称>  删除数据库的语句格式：  DROP DATABASE <数据库名称>  注意：在删除数据库之前，先关闭要删除的数据库。方法是使用USE打开另外一个数据库。  **二、实验任务**  1.打开数据库SQL Server 2008的查询分析器，用Create Table建表Student，course，sc。数据库的名称为“学生选课”，各个表结构如教材P79页所示。  以下要求用T-SQL来实现：  CREATEDATABASE学生选课  ON  (NAME='学生选课\_dat',  FILENAME='D:\data\学生选课\_dat.mdf')  LOGON  (NAME='学生选课\_Log',  FILENAME='D:\data\学生选课\_Log.ldf ')  USE学生选课  CREATETABLE Student(  Sno CHAR(9) PRIMARYKEY,  Sname CHAR(20) UNIQUE, SsexCHAR(2),  Sage SMALLINT,  Sdept CHAR(20)  );  CREATETABLE Course  (Cno CHAR(4) PRIMARYKEY,  Cname CHAR(40),  Cpno CHAR(4), CcreditSMALLINT,  FOREIGN KEY (Cpno) REFERENCES Course(Cno)  );  CREATE TABLE SC  (SNo CHAR(9),  CNo CHAR(4),  Grade SMALLINT,  PRIMARY KEY (SNo,CNo),  FOREIGN KEY (Sno) REFERENCES Student(Sno),  FOREIGNKEY (Cno)REFERENCES Course(Cno));     1. 向表中填入教材P79数据。   INSERT INTO Student VALUES('200215121','李勇','男',20,'CS');  INSERT INTO Student VALUES('200215122','刘晨','女',19,'CS');  INSERT INTO Student VALUES('200215123','王敏','女',18,'MA');  INSERT INTO Student VALUES('200215125','张立','男',19,'IS');  INSERT INTO Course VALUES('1','数据库','5',4);  INSERT INTO Course VALUES('2','数学', null,2);  INSERT INTO Course VALUES('3','信息系统','1',4);  INSERT INTO Course VALUES('4','操作系统','6',3);  INSERT INTO Course VALUES('5','数据结构','7',4);  INSERT INTO Course VALUES('6','数据处理',null,2);  INSERT INTO Course VALUES('7','PASCAL语言','6',4);  INSERT INTO SC VALUES('200215121','1',92);  INSERT INTO SC VALUES('200215121','2',85);  INSERT INTO SC VALUES('200215121','3',88);  INSERT INTO SC VALUES('200215122','2',90);  INSERT INTO SC VALUES('200215122','3',80);     1. 向Student 表增加“入学时间”列（列名为Scome,日期型）   ALTER TABLE Student ADD Scome DATETIME     1. 将年龄的数据类型改为整型。   ALTER TABLE Student ALTER COLUMN Sage int     1. 为Student中sname添加列级完整性约束，不能为空。为Student中sno添加列级完整性约束，取值唯一，不能为空。   ALTER TABLE Student ADD CONSTRAINT C1 CHECK (Sname is not null);  ALTER TABLE Student ADD CONSTRAINT C2 CHECK (Sno is not null);    ALTER TABLE Student ADD UNIQUE(Sno)     1. 删除Student中sname列级完整性约束。   ALTER TABLE Student DROP CONSTRAINT C1;     1. 为SC建立按学号升序和课程号降序建立唯一索引。   ALTER TABLE Student DROP CONSTRAINT C1;  CREATE UNIQUE INDEX SCno ON SC (Sno ASC,Cno DESC);     1. 建立Student、Course及SC的主码约束   ALTER TABLE Student ADD CONSTRAINT C3 primary key (Sno)  ALTER TABLE Course ADD CONSTRAINT C4 primary key (Cno)  ALTER TABLE sc ADD CONSTRAINT C5 primary key (Sno,Cno)     1. 建立 Student与SC、Course与SC之间的外码约束   Alter table sc add constraint c3 foreign key (sno) references student (sno);  Alter table sc add constraint c4 foreign key (cno) references course (cno);     1. 实现学生性别只能是“男”或“女”的Check(检查)约束。   ALTER TABLE Student ADD CONSTRAINT C5 CHECK(Ssex in ('男','女'))     1. 实现学生年龄只能小于等于30大于17的语义约束   ALTER TABLE Student ADD CONSTRAINT C6 CHECK (Sage>17 and Sage<=30);     1. 用Create Index对表student的sname字段建立一个升序索引，索引名Indexs。   create index indexs on student (sname asc)     1. 用Drop Index删除索引Indexs。   DROP INDEX INDEXS ON Student     1. SC表的成绩值应该在0到100之间。   Alter table SC add CHECk(Grade between 0 and 100);     1. 修改表Student中的约束条件，要求学号在900000~999999之间，年龄小于40。   Alter table student add CHECk(SNo between 90000000 and 999999999 and SAge < 40);     1. 限制数据库课程最多60名学生选修。   IF (OBJECT\_ID('TGR\_SC\_INSERT', 'TR') IS NOT NULL)  DROP TRIGGER TGR\_SC\_INSERT  GO  CREATE TRIGGER TGR\_SC\_INSERT  ON SC  FOR INSERT  AS  DECLARE @COUNT INT,@Sno CHAR(9),@Cno CHAR(4), @Grade SMALLINT;  SELECT @COUNT=COUNT(\*) FROM Course,SC WHERE SC.Cno=Course.Cno AND Course.Cname='数据库';  SELECT @Sno=Sno,@Cno=Cno,@Grade=Grade FROM INSERTED; IF(@COUNT>3)  BEGIN  DELETE SC WHERE Sno=@Sno AND Cno=@Cno AND Grade=@Grade;  END  GO     1. 限制每一门课程最多60名学生选修。   IF (OBJECT\_ID('TGR\_SC\_INSERT', 'TR') IS NOT NULL)  DROP TRIGGER TGR\_SC\_INSERT  GO  CREATE TRIGGER TGR\_SC\_INSERT  ON SC  FOR INSERT  AS    DECLARE @COUNT INT,@Sno CHAR(9),@Cno CHAR(4), @Grade SMALLINT;  SELECT count(\*) FROM SC GROUP by CNo)  SELECT @COUNT=COUNT(\*) FROM Course,SC WHERE SC.Cno=Course.Cno AND Course.Cname='数据库';  SELECT @Sno=Sno,@Cno=Cno,@Grade=Grade FROM INSERTED; -- inserted  IF(@COUNT>60)  BEGIN  DELETE SC WHERE Sno=@Sno AND Cno=@Cno AND Grade=@Grade;  END  GO     1. 限制每学期每一门课程最多60名学生选修。   IF (OBJECT\_ID('TGR\_SC\_INSERT', 'TR') IS NOT NULL)  DROP TRIGGER TGR\_SC\_INSERT  GO  CREATE TRIGGER TGR\_SC\_INSERT  ON SC  FOR INSERT  AS    DECLARE @COUNT INT,@Sno CHAR(9),@Cno CHAR(4), @Grade SMALLINT;  SELECT count(\*) FROM SC GROUP by CNo，term)  SELECT @COUNT=COUNT(\*) FROM Course,SC WHERE SC.Cno=Course.Cno AND Course.Cname='数据库';  SELECT @Sno=Sno,@Cno=Cno,@Grade=Grade FROM INSERTED; -- inserted  IF(@COUNT>60)  BEGIN  DELETE SC WHERE Sno=@Sno AND Cno=@Cno AND Grade=@Grade;  END  GO    19.为Course表的Cname列增加一个唯一性约束。  alter table course add constraint cname\_unique unique(cname);     1. 删除Course表的Cname列唯一性约束。   ALTER TABLE course DROP CONSTRAINT cname\_unique | | | | |
|  | | | | | |