**《数据库原理》实验指导书**

**2018年4月 第一版**

**2020年9月 修订**

**内容简介**

数据库原理实验是本《数据库原理》课程的重要实践教学环节。实验的目的不仅仅是验证理论知识，更重要的是通过实验加强学生的实验手段与实践技能，掌握使用实际数据库系统的使用方法，加深理论知识的理解，培养学生分析问题、解决问题、应用知识的能力和创新精神，真正培养学生的实践动手能力，全面提高学生的综合素质。

数据库原理实验在客户/服务器环境下的客户终端计算机上完成，学生通过客户端远程连接数据库服务器。本实验指导书以PosgreSQL+Windows作为数据库服务器实验环境，学生可以根据个人兴趣选择其他数据库管理系统和操作系统平台。实验中每人一台微机，前两个实验是一人一组，第三、四个实验为两人一组，也可每人在自己本地服务器创建两个用户实现。

实验基本要求：每位学生能独立完成实验内容，还能进行简单的错误分析和错误排除，完成课下习题。

实验目的：通过实验让学生掌握数据库系统的安装、数据库的创建、SQL语句的应用，加深数据库管理系统中并发控制、完整性和安全机制的理论理解，增强实际应用技能。

# 实验1: 数据库管理系统安装与数据库定义

1. **实验目的：**
   1. 通过对数据库管理系统的安装使用，初步了解DBMS的工作环境和系统构架。
   2. 熟悉对DBMS的安装，搭建今后实验的平台
   3. 创建数据库，在数据库上建立关系模式，插入数据，进行相应的查询操作。

**二、实验要求**

1. 了解当前主流关系数据库系统，了解RDBMS的系统框架，如客户/服务器结构。
2. 预习数据库系统安装过程。
3. 初步了解RDBMS的基础安全性。例如，用户登陆和创建系统管理员。
4. 能够基于数据库设计，创建数据库。
5. 能够对定义数据库完整性。
6. 能够对数据库进行备份和恢复。

**三、实验内容**

（一）熟悉数据库管理系统环境。

1.实验目的

了解主流关系数据库管理系统

* Oracle
* Microsoft SQL Server 2016
* MYSQL
* PostgreSQL
* DB2
* Microsoft Access

图1为DB-Engines2020年9月发布的数据库流行度排名。实时的排名信息可以访问<https://db-engines.com/en/ranking>

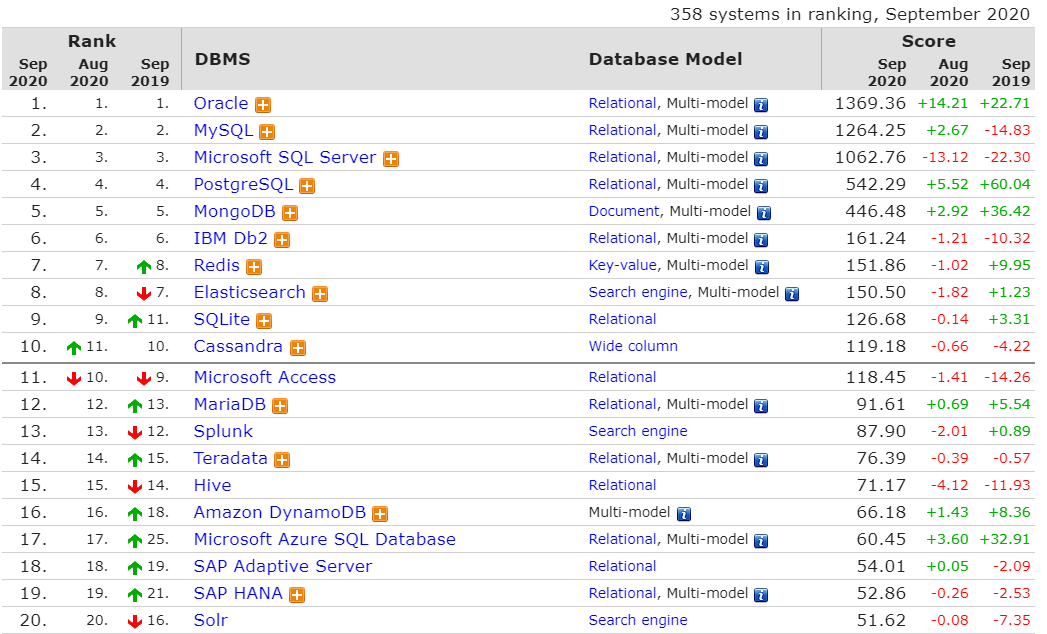


图1 数据库流行度

2.实验内容

安装数据库管理系统

当前数据库管理系统均采用客户/服务器结构，学生可尝试在不同的数据库管理系统与操作系统环境安装数据库管理系统的服务器端和客户端。

（1）PostgreSQL+ Windows（推荐）

数据库管理系统软件: PostgreSQL 10.X （32位或64位）

服务器端操作系统：Windows7,8,10，Windows server 2008/2012/2016

完成方式：课前或课堂完成

安装所需文件：postgresql-10.X-1-windows.exe（包含PostgreSQL客户端软件）

操作步骤如下：

1) 双击安装文件，选择安装路径；

2）设置数据库的超级用户postgres的密码；

3）设置数据库服务监听端口，默认为5432

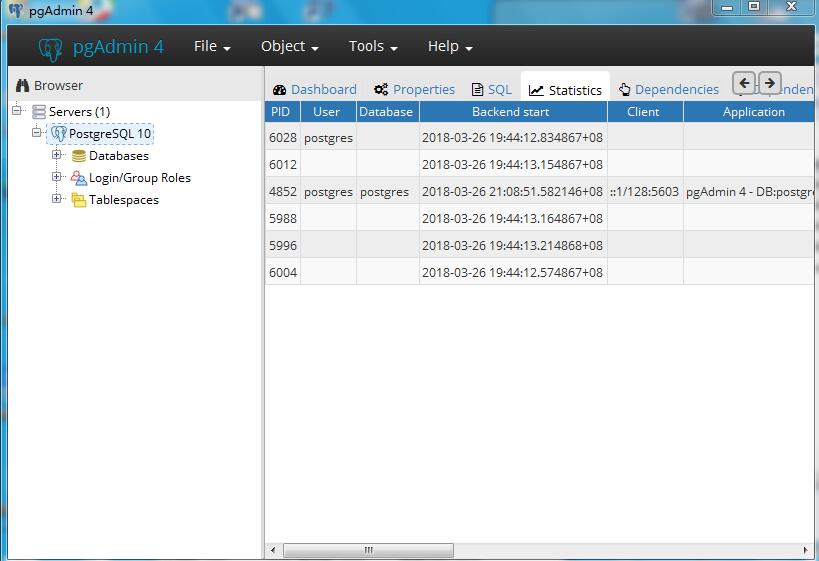
4）选择服务器集群，这里用默认设置default locale就可以；

5）点击next直至完成安装。

安装后可以在系统的服务中看到postgresql的服务，如下图



使用postgreSQL的Windows客户端软件pgAdmin4.X登录查看



（2）PostgreSQL源码+ Linux（推荐对数据库深入学习的同学）

数据库管理系统软件: PostgreSQL 9.X （64位）

服务器端操作系统：linux（centOS/Ubuntu）

完成方式：课后自愿

（3）SQL Server + Windows

数据库管理系统软件: SQL Server 2016 Express版本

服务器端操作系统：Windows7,8,10，Windows server 2008/2012

完成方式：课后自愿完成

（4）MYSQL+ Windows

数据库管理系统软件: MYSQL版本

服务器端操作系统：Windows7,8,10，Windows server 2008/2012

完成方式：课后自愿完成

（5）Oracle+ Windows

数据库管理系统软件: Oracle12

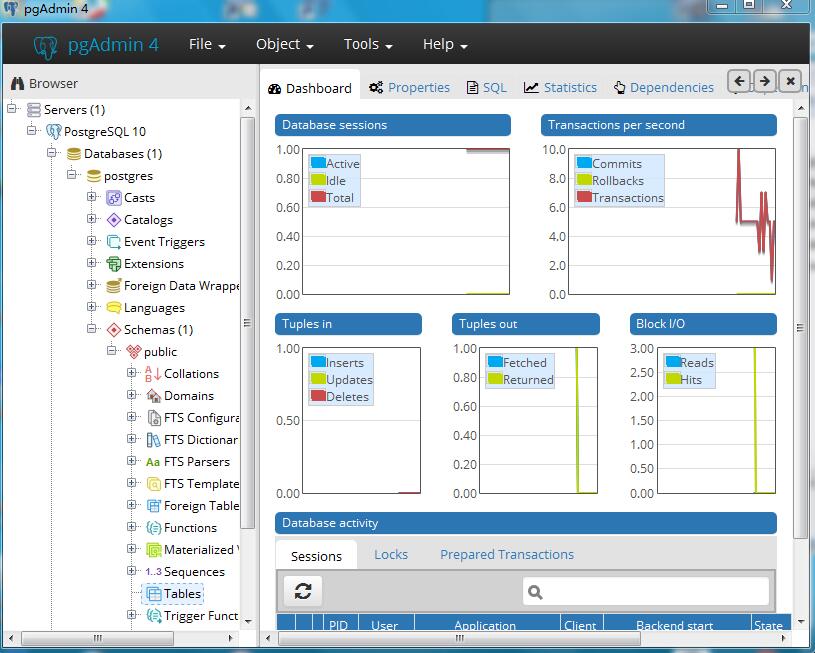
服务器端操作系统：Windows7,8,10，Windows server 2008/2012

完成方式：课后自愿完成

同学可以根据自己的兴趣与能力尝试其他数据库管理系统与操作系统组合。

3.关系数据库管理系统的基本操作

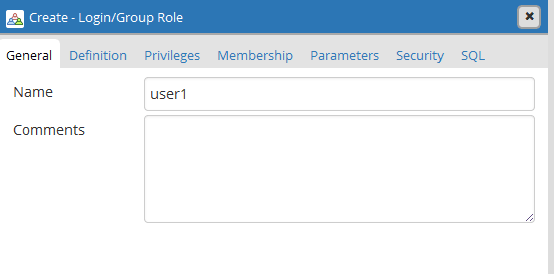
（1）PostgreSQL



1）了解PostgreSQL客户端。

启动PostgreSQL的客户端pgAdmin4，左侧为树形数据库菜单，依次为服务器（servers）、数据库（databases）。在servers中可以选择本机服务器或添加远程服务器。上图中默认数据库为postgres，数据表在schemas-public-tables中。

数据库用户在Login/Group Roles目录中，数据库初始只有系统用户postgres。可以点击右键通过菜单创建新用户。创建用户菜单如下图，其中general中填入用户名，Definition中设置密码，privilege中设置权限。创建“user1”用户，并设置login、create database权限。



客户端右侧选项卡“dashboard”为系统运行监控信息，properties为所选对象的属性信息，“SQL”为查看SQL语句的界面。

（2）MYSQL

课后自学

（3）MS SQL Server

课后自学

4.实验报告

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 实验报告1.1 | | | | |
| 题目：数据库原理实验环境安装 | 姓名 |  | 班级 |  |
| 学号 |  | 时间 |  |
| 1． 数据库管理系统安装  数据库原理实验所使用的软件环境（可自行选择在自己机器上安装）   |  |  | | --- | --- | | 操作系统： |  | | 数据库管理系统： |  | | 数据库管理系统版本： |  |   2． 实验环境搭建中遇到了什么问题？如何解决的？  3．数据库管理系统运行的操作系统环境。查找资料，了解各数据库管理系统对操作系统平台的支持情况，完成下表。   |  |  |  |  |  | | --- | --- | --- | --- | --- | | RDBMS | 最新版本号 | Windows | Linux（centos/ubuntu） | Mac OS | | Oracle |  |  |  |  | | SQL Server |  |  |  |  | | MYSQL |  |  |  |  | | PostgreSQL |  |  |  |  | | DB2 |  |  |  |  |   4\*. 了解各数据库管理系统的客户机服务器机制，完成下表：   |  |  |  | | --- | --- | --- | | RDBMS | 客户端软件名称 | 服务器端进程名 | | Oracle |  |  | | SQL Server |  |  | | MYSQL |  |  | | PostgreSQL |  |  |   5\*. 查询相关资料，说明PostgreSQL与MYSQL的主要差异？（选做） | | | | |
| 实验总结：  1 实验中遇到的难点有哪些？  2 实验自我评价。 | | | | |

5.重要链接

PostgreSQL:

https://www.postgresql.org/

PostgreSQL安装程序

https://www.enterprisedb.com/downloads/postgres-postgresql-downloads

PgAdmin安装程序:

https://www.pgadmin.org/download/

（二）数据库定义与基本操作（1）

1. 实验目的

本实验的目的是理解和掌握数据库的定义语言，使用SQL DDL语言创建、修改和删除数据库的模式和关系表。

2. 实验内容

使用SQL DDL语句创建数据库；

联系基于ER图设计数据库的关系表结构；

使用SQL DDL在数据库中创建基本表；

使用SQL进行基本的数据操作；

验证数据库的实体完整性约束。

3. 实验准备知识

本实验基于学生选修课程数据库进行，描述学生、课程情况的E-R图如下：

4. 实验报告

学生

课程

选修

学号

姓名

年龄

所在系

成绩

课号

课程名

学时

学分

先行课号

m

n

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 实验报告1.2 | | | | |
| 题目：数据库定义与基本操作 | 姓名 |  | 班级 |  |
| 学号 |  | 时间 |  |
| 1. 请将ER图转化为用关系数据模型描述的关系模式。 2. 创建学生选课数据库Enrolled   使用实验1.1中方法创建用户user1，并用该用户创建数据库“Enrolled”。 可以使用SQL语句和图形界面  1）方法1：使用SQL语句  点击菜单中的Tool->Query Tool，输入以下SQL语句并执行  CREATE DATABASE "Enrolled"  WITH  OWNER = user1  ENCODING = 'UTF8'  CONNECTION LIMIT = -1;  2）方法2：使用图形界面  右键点击左侧“database”目录，选择其中create->database，出现下图界面，在General选项卡中填写数据库名称（“database”）、用户（“Owner”），在Definition选项卡中可以定义数据库所使用的字符编码方式（“Encoding”）、表空间（“tablespace”），允许的连接数量“connection limit”等设置，最后点击“save”创建。创建后可以在左侧列表的“database”目录中查看数据库“Enrolled”。     1. 定义数据库基本表   1）关系基本表定义语句  在 PostgreSQL中创建关系表同样可以使用SQL和图形界面两种方式，这里我们介绍使用SQL语句方法。右键点击“Enrolled”数据库，选择其中Query Tool打开查询界面。  学生关系表：  CREATE TABLE Student (  Sno VARCHAR (20),  Sname VARCHAR(30),  Ssex VARCHAR (2),  Sage INTEGER,  Sdept VARCHAR (30),  PRIMARY KEY (Sno))  课程关系表：  CREATE TABLE Course (  Cno VARCHAR (10) not null,  Cname VARCHAR (30),  Chour INTEGER,  Credit VARCHAR (30),  Cpno VARCHAR (10)  )  执行结果如何？  2)实体完整性约束  i 具有主键的关系表  insert into Student values('95020', '陈东', 'M', 18,'CS')  insert into Student values('95020', '陈东', 'M', 18,'CS')  插入以上两条记录数据，依次执行上面的SQL语句，执行结果如何？为什么？  ii 未声明主键的关系表  insert into Course values('c1', '数据库原理', 40,2,null)  insert into Course values('c1', '数据库原理', 40,2,null)  插入以上两条记录数据，依次执行上面的SQL语句，次执行结果如何？为什么？  3）定义完整性约束  在已建立的关系模式之上（已插入一些数据）建立主键约束，参照约束和用户定义的约束（要求学生年龄不小于14岁，不大于35岁），如果约束不能建立，请分析原因，修改后建立上述约束。  i 定义主键约束  执行以下语句，返回结果如何？为什么？  alter table Course add constraint pk\_courses primary key(Cno)  如何可以解决定义主键的问题？  delete from Course  where Cno='c1'  约束建立后，请试着利用如下语句插入重复元组，结果如何？为什么？  insert into Course values('c1', '数据库原理', 40,2, null)  insert into Course values('c1', '数据库原理', 40,2, null)  ii 创建用户自定义约束  执行以下语句  alter table Student add constraint u\_Student check (Sage between 14 and 35)  执行以下语句，结果如何？为什么？  insert into Student values('95021', '王二', 'M', 12,'IS')  4）关系模式定义语句联系  编写创建选课关系表SC（Sno,Cno,Grade），要求添加主键和外键，相关SQL语句如何编写？  create table SC (  Sno VARCHAR (20),  Cno VARCHAR(10),  Grade integer,  primary key (Sno, Cno),  FOREIGN KEY (Cno) REFERENCES Course (Cno),  FOREIGN KEY (Sno) REFERENCES Student (Sno)) | | | | |
| 实验总结：  1 实验中遇到的难点有哪些？  2 实验自我评价。 | | | | |

（三）数据库定义与基本操作（2）

1. 实验目的

本实验的目的是练习使用SQL DDL语言进行数据库测试数据集的创建，使用SQL DDL语言创建数据库测试数据集的关系表，能够向数据库中实现数据的导入。

2. 实验内容

了解TPC-H基准测试

使用SQL DDL语句创建TPC-H数据库；

对TPC-H数据库添加生成好的测试数据。

3. 实验准备知识

（1）**熟悉TPC-H基准测试**

事务处理性能委员会（ Transaction ProcessingPerformance Council ），是由数10家会员公司创建的非盈利组织，总部设在美国。TPC- H 主要目的是评价特定查询的决策支持能力，强调服务器在数据挖掘、分析处理方面的能力。查询是决策支持应用的最主要应用之一，数据仓库中的复杂查询可以分成两种类型：一种是预先知道的查询，如定期的业务报表；另一种则是事先未知的查询，称为动态查询（Ad- Hoc Query）。TPC-H就是当一家数据库开发商开发了一个新的数据库操作系统，采用TPC-H作为测试基准，来测试衡量数据库操作系统查询决策支持方面的能力．它模拟决策支持系统中的数据库操作，测试数据库系统复杂查询的响应时间，以每小时执行的查询数(TPC-H QphH@Size)作为度量指标。数据库模型见图2，共有8 张表，除Nation 和Region 表外，其它表与测试的数据量有关，即比例因SF（Scale Factor）



图1 TPC成员企业

在本课程的实验中，将使用TPC-H基准测试作为实验的数据模式，并在涉及查询性能和查询计划的相关实验中使用基于其模式生成的数据进行实验。



图1 TPC-H数据模式

4. 实验报告

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 实验报告1.3 | | | | |
| 题目：TPC-H数据库创建于数据导入 | 姓名 |  | 班级 |  |
| 学号 |  | 时间 |  |
| 1创建TPC-H数据库  参考步骤：  （1）在PostgreSQL中创建数据库TPC  使用具有创建数据库权限的用户登录，创建数据库  CREATE DATABASE TPC；  （2）在数据库TPC中创建schema  创建模式TPCH  CREATE SCHEMA TPCH；  注：schema的作用：   * 方便管理多个用户共享一个数据库,但是又可以互相独立. * 方便管理众多对象,更有逻辑性 * 方便兼容某些第三方应用程序,创建对象时是有schema的   比如要设计一个复杂系统.由众多模块构成,有时候模块间又需要有独立性.各模块存放单独的数据库显然是不合适的。这时候使用schema来分类各模块间的对象,再对用户进行适当的权限控制.这样逻辑也非常清晰。  2创建TPC-H基准测试的关系表  基于TPC-H数据模式，结合以下表结构信息，编写建表SQL语句，创建关系表  **(1)** 零件**PART Table**   |  |  |  | | --- | --- | --- | | Column Name | Data Type | Comment | | P\_PARTKEY | identifier | SF\*200,000 | | P\_NAME | variable text, size 55 |  | | P\_MFGR | fixed text, size 25 |  | | P\_BRAND | fixed text, size 10 |  | | P\_TYPE | variable text, size 25 |  | | P\_SIZE | integer |  | | P\_CONTAINER | fixed text, size 10 |  | | P\_RETAILPRICE | decimal |  | | P\_COMMENT | variable text, size 23 |  | | PRIMARY KEY: P\_PARTKEY | | |   **(2)** 供应商**SUPPLIER Table Layout**   |  |  |  | | --- | --- | --- | | Column Name | Data Type | Comment | | S\_SUPPKEY | identifier | SF\*10,000 | | S\_NAME | fixed text, size 25 |  | | S\_ADDRESS | variable text, size 40 |  | | S\_NATIONKEY | Identifier | Foreign Key to N\_NATIONKEY | | S\_PHONE | fixed text, size 15 |  | | S\_ACCTBAL | decimal |  | | S\_COMMENT | variable text, size 101 |  | | PRIMARY KEY: S\_SUPPKEY | | |   **(3) 零件供应PARTSUPP Table Layout**   |  |  |  | | --- | --- | --- | | Column Name | Data Type | Comment | | PS\_PARTKEY | identifier | Foreign Key to P\_PARTKEY | | PS\_SUPPKEY | Identifier | Foreign Key to S\_SUPPKEY | | PS\_AVAILQTY | integer |  | | PS\_SUPPLYCOST | Decimal |  | | PS\_COMMENT | variable text, size 199 |  | | PRIMARY KEY: PS\_PARTKEY，PS\_SUPPKEY | | |   **(4) 客户CUSTOMER Table Layout**   |  |  |  | | --- | --- | --- | | Column Name | Data Type | Comment | | C\_CUSTKEY | identifier | SF\*150,000 | | C\_NAME | variable text, size 25 |  | | C\_ADDRESS | variable text, size 40 |  | | C\_NATIONKEY | Identifier | Foreign Key to N\_NATIONKEY | | C\_PHONE | fixed text, size 15 |  | | C\_ACCTBAL | decimal |  | | C\_MKTSEGMENT | fixed text, size 10 |  | | C\_COMMENT | variable text, size 117 |  | | PRIMARY KEY: C\_CUSTKEY | | |   **(5) 订单ORDERS Table Layout**   |  |  |  | | --- | --- | --- | | Column Name | Data Type | Comment | | O\_ORDERKEY | identifier | SF\*1,500,000 | | O\_CUSTKEY | identifier | Foreign Key to C\_CUSTKEY | | O\_ORDERSTATUS | fixed text, size 1 |  | | O\_TOTALPRICE | Decimal |  | | O\_ORDERDATE | Date |  | | O\_ORDERPRIORITY | fixed text, size 15 |  | | O\_CLERK | fixed text, size 15 |  | | O\_SHIPPRIORITY | Integer |  | | O\_COMMENT | variable text, size 79 |  | | PRIMARY KEY: O\_ORDERKEY | | |   备注：Orders are not present for all customers. In fact, one-third of the customers do not have any order in the database. The orders are assigned at random to two-thirds of the customers (see Clause 4: ). The purpose of this is to exercise the capabilities of the DBMS to handle "dead data" when joining two or more tables  **(6) 明细支出 LINEITEM Table Layout**   |  |  |  | | --- | --- | --- | | Column Name | Data Type | Comment | | L\_ORDERKEY | identifier | Foreign Key to O\_ORDERKEY | | L\_PARTKEY | identifier | Foreign key to P\_PARTKEY, first part of the compound Foreign Key to (PS\_PARTKEY, PS\_SUPPKEY) with L\_SUPPKEY | | L\_SUPPKEY | identifier | Foreign key to S\_SUPPKEY, second part of the compound Foreign Key to (PS\_PARTKEY,  PS\_SUPPKEY) with L\_PARTKEY | | L\_LINENUMBER | integer |  | | L\_QUANTITY | decimal |  | | L\_EXTENDEDPRICE | decimal |  | | L\_DISCOUNT | decimal |  | | L\_TAX | decimal |  | | L\_RETURNFLAG | fixed text, size 1 |  | | L\_LINESTATUS | fixed text, size 1 |  | | L\_SHIPDATE | date |  | | L\_COMMITDATE | date |  | | L\_RECEIPTDATE | date |  | | L\_SHIPINSTRUCT | fixed text, size 25 |  | | L\_SHIPMODE | fixed text, size 10 |  | | L\_COMMENT | variable text size 44 |  | | Primary Key**:** L\_ORDERKEY, L\_LINENUMBER | | |   **(7) 国家NATION Table Layout**   |  |  |  | | --- | --- | --- | | Column Name | Data Type | Comment | | N\_NATIONKEY | identifier | 25 nations are populated | | N\_NAME | fixed text, size 25 | Foreign Key to R\_REGIONKEY | | N\_REGIONKEY | identifier |  | | N\_COMMENT | variable text, size 152 |  | | Primary Key**:** N\_NATIONKEY | | |   **(8)地区 REGION Table Layout**   |  |  |  | | --- | --- | --- | | Column Name | Data Type | Comment | | R\_REGIONKEY | identifier | 5 regions are populated | | R \_NAME | fixed text, size 25 |  | | R \_COMMENT | variable text, size 152 |  | | Primary Key**:**  R\_REGIONKEY |  |  | |  | | |   编写SQL DDL语句，创建以上数据库关系表：  (1) PART  CREATE TABLE PART  (P\_PARTKEY INTEGER,  P\_NAME VARCHAR (55),  P\_MFGR CHAR(25),  P\_BRAND CHAR (10),  P\_TYPE VARCHAR (25),  P\_SIZE INTEGER,  P\_CONTAINER CHAR (10),  P\_RETAILPRICE DECIMAL (15,2),  P\_COMMENT VARCHAR (23),  PRIMARY KEY(P\_PARTKEY));  (2) **SUPPLIER**  CREATE TABLE SUPPLIER (  S\_SUPPKEY INTEGER,  S\_NAME CHAR (25),  S\_ADDRESS VARCHAR (40),  S\_NATIONKEY INTEGER,  S\_PHONE CHAR (15),  S\_ACCTBAL DECIMAL (15,2),  S\_COMMENT VARCHAR (101),  PRIMARY KEY(S\_SUPPKEY));  (3) **PARTSUPP**  CREATE TABLE PARTSUPP (  PS\_PARTKEY INTEGER REFERENCES PART(P\_PARTKEY),  PS\_SUPPKEY INTEGER REFERENCES SUPPLIER (S\_SUPPKEY),  PS\_AVAILQTY INTEGER,  PS\_SUPPLYCOST DECIMAL (15,2),  PS\_COMMENT VARCHAR (199),  PRIMARY KEY (PS\_PARTKEY, PS\_SUPPKEY));  (4) **CUSTOMER**  CREATE TABLE CUSTOMER (  C\_CUSTKEY INTEGER PRIMARY KEY,  C\_NAME VARCHAR (25),  C\_ADDRESS VARCHAR (40),  C\_NATIONKEY INTEGER REFERENCES NATION (N\_NATIONKEY),  C\_PHONE CHAR (15),  C\_ACCTBAL DECIMAL (15,2),  C\_MKTSEGMENT CHAR (10),  C\_COMMENT VARCHAR (117));  (5) **ORDERS**  CREATE TABLE ORDERS (  O\_ORDERKEY INTEGER PRIMARY KEY,  O\_CUSTKEY INTEGER REFERENCES CUSTOMER (C\_CUSTKEY),  O\_ORDERSTATUS CHAR (1),  O\_TOTALPRICE DECIMAL (15,2),  O\_ORDERDATE DATE,  O\_ORDERPRIORITY CHAR (15),  O\_CLERK CHAR (15),  O\_SHIPPRIORITY INTEGER,  O\_COMMENT VARCHAR (79));  (6) **LINEITEM**  CREATE TABLE LINEITEM (  L\_ORDERKEY INTEGER REFERENCES ORDERS(O\_ORDERKEY),  L\_PARTKEY INTEGER REFERENCES PART(P\_PARTKEY),  L\_SUPPKEY INTEGER REFERENCES SUPPLIER (S\_SUPPKEY),  L\_LINENUMBER INTEGER,  L\_QUANTITY DECIMAL (15,2),  L\_EXTENDEDPRICE DECIMAL (15,2),  L\_DISCOUNT DECIMAL (15,2),  L\_TAX DECIMAL (15,2),  L\_RETURNFLAG CHAR (1),  L\_LINESTATUS CHAR (1),  L\_SHIPDATE DATE,  L\_COMMITDATE DATE,  L\_RECEIPTDATE DATE,  L\_SHIPINSTRUCT CHAR (25),  L\_SHIPMODE CHAR(10),  L\_COMMENT VARCHAR (44),  PRIMARY KEY (L\_ORDERKEY, L\_LINENUMBER),  FOREIGN KEY (L\_PARTKEY, L\_SUPPKEY) REFERENCES PARTSUPP (PS\_PARTKEY, PS\_SUPPKEY));  (7) NATION  CREATE TABLE NATION (  N\_NATIONKEY INTEGER PRIMARY KEY,  N\_NAME CHAR (25),  N\_REGIONKEY INTEGER REFERENCES REGION (R\_REGIONKEY),  N\_COMMENT VARCHAR (152));  (8) REGION  CREATE TABLE REGION (  R\_REGIONKEY INTEGER PRIMARY KEY,  R\_NAME CHAR (25),  R\_COMMENT VARCHAR (152));  以上建表语句为包含完整性约束建表语句。在进行数据导入时，完整性约束将导致数据导入的性能下降。因此，请将以上建表语句中涉及实体完整性约束和参照完整性约束的部分删除，并在数据库的模式TPCH中创建关系表。  请给出删除完整性约束后的SQL语句：  3导入已生成的TPC-H基准测试数据  将使用工具自动生成的TPC-H基准测试数据导入到数据库中，具体操作步骤如下：  （1）将当前使用的数据库设置为TPC；  （2）使用Query Tool导入数据：  数据打包为tables-out.rar文件，请将文件解压后存储在本地路径；  使用Query Tool执行如下脚本导入数据，注意替换其中红色的路径为本地数据文件路径。  D:\Software\Protage11.2\table-out-data\tables-out  Copy region FROM 'F:/Data/tables/tables-out/region.tbl' WITH DELIMITER AS '|';  Copy nation FROM 'F:/ Data/tables/tables-out/nation.tbl' WITH DELIMITER AS '|';  Copy part FROM 'F:/ Data/tables/tables-out/part.tbl' WITH DELIMITER AS '|';  Copy supplier FROM 'F:/ Data/tables/tables-out/supplier.tbl' WITH DELIMITER AS '|';  Copy customer FROM 'F:/ Data/tables/tables-out/customer.tbl' WITH DELIMITER AS '|';  Copy lineitem FROM 'F:/ Data/tables/tables-out/lineitem.tbl' WITH DELIMITER AS '|';  Copy partsupp FROM 'F:/ Data/tables/tables-out/partsupp.tbl' WITH DELIMITER AS '|';  Copy orders FROM 'F:/ Data/tables/tables-out/orders.tbl' WITH DELIMITER AS '|';  注意：Copy命令要求数据文件在服务器本地  4添加约束  数据导入后，向数据库增加数据完整性约束。执行以下脚本：  -- For table REGION  ALTER TABLE REGION  ADD PRIMARY KEY (R\_REGIONKEY);  -- For table NATION  ALTER TABLE NATION  ADD PRIMARY KEY (N\_NATIONKEY);  ALTER TABLE NATION  ADD FOREIGN KEY (N\_REGIONKEY) references REGION;  COMMIT WORK;  -- For table PART  ALTER TABLE PART  ADD PRIMARY KEY (P\_PARTKEY);  COMMIT WORK;  -- For table SUPPLIER  ALTER TABLE SUPPLIER  ADD PRIMARY KEY (S\_SUPPKEY);  ALTER TABLE SUPPLIER  ADD FOREIGN KEY (S\_NATIONKEY) references NATION;  COMMIT WORK;  -- For table PARTSUPP  ALTER TABLE PARTSUPP  ADD PRIMARY KEY (PS\_PARTKEY, PS\_SUPPKEY);  COMMIT WORK;  -- For table CUSTOMER  ALTER TABLE CUSTOMER  ADD PRIMARY KEY (C\_CUSTKEY);  ALTER TABLE CUSTOMER  ADD FOREIGN KEY (C\_NATIONKEY) references NATION;  COMMIT WORK;  -- For table LINEITEM  ALTER TABLE LINEITEM  ADD PRIMARY KEY (L\_ORDERKEY, L\_LINENUMBER);  COMMIT WORK;  -- For table ORDERS  ALTER TABLE ORDERS  ADD PRIMARY KEY (O\_ORDERKEY);  COMMIT WORK;  -- For table PARTSUPP  ALTER TABLE PARTSUPP  ADD FOREIGN KEY (PS\_SUPPKEY) references SUPPLIER;  COMMIT WORK;  ALTER TABLE PARTSUPP  ADD FOREIGN KEY (PS\_PARTKEY) references PART;  COMMIT WORK;  -- For table ORDERS  ALTER TABLE ORDERS  ADD FOREIGN KEY (O\_CUSTKEY) references CUSTOMER;  COMMIT WORK;  -- For table LINEITEM  ALTER TABLE LINEITEM  ADD FOREIGN KEY (L\_ORDERKEY) references ORDERS;  COMMIT WORK;  ALTER TABLE LINEITEM  ADD FOREIGN KEY (L\_PARTKEY, L\_SUPPKEY) references PARTSUPP;  COMMIT WORK;  以上脚本执行顺序是否可以改变？为什么？  3 执行基本查询  对数据库各个表进行查询测试，查看其中数据 | | | | |
| 实验总结：  1 TPC-H的ER图应该是什么样的？请还原出TPC-H的ER图  2 在TPCH的数据导入环节中，关系表数据的导入顺序是否可以改变？如果在建表时增加了完整性约束，数据导入的顺序是否可以改变？如果需要改变，应如何调整，为什么？  3 实验中遇到的难点有哪些？  4 实验自我评价。 | | | | |

（四）数据库备份与还原

1. 实验目的

本实验的目的是练习使用PostgreSQL的客户端工具对数据库进行备份和恢复。

2. 实验内容

了解pgAdmin工具

使用pgAdmin的数据备份和恢复功能。

3. 实验准备知识

4 实验报告

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 实验报告1.4 | | | | |
| 题目：PostgreSQL数据库备份与恢复操作 | 姓名 |  | 班级 |  |
| 学号 |  | 时间 |  |
| 1 PostgreSQL数据库备份  （1）在pgAdmin界面中左侧树状菜单中选择要备份的数据库  （2）右键点击数据库，出现菜单    （3）选择backup选项，填写相关信息    （4）点击“backup”完成备份  2 PostgreSQL数据库恢复  （1）在pgAdmin中创建要恢复的数据库  （2）在数据库右键菜单中选择“restore”，选择数据库备份文件进行恢复  3 每次实验课课后备份数据库，并复制文件，在下次实验课时恢复数据库并在恢复的数据库上进行实验操作。 | | | | |
| 实验总结：  1 实验中遇到的难点有哪些？  2 实验自我评价。 | | | | |

**四、课后练习（选做）**

1. 练习安装和操作Oracle、MYSQL和SQL Server数据库管理系统。

2. 练习使用TPC-H基准测试程序生成测试数据集。

**实验报告格式**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 实验报告X.X | | | | |
| 题目： | 姓名 |  | 班级 |  |
| 学号 |  | 时间 |  |
|  | | | | |
| 实验总结：  1 实验中遇到的难点有哪些？  2 实验自我评价。 | | | | |