

**实 验 报 告**

**（ 2024/2025 学年 第 一 学期）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | 数据库系统原理 | | | | | |
| 实验名称 | 约束、视图、索引与存储过程 | | | | | |
| 实验时间 | 2024 | 年 | 12 | 月 | 9 | 日 |
| 指导单位 | 物联网学院 | | | | | |
| 指导教师 |  | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 学生姓名 |  | 班级学号 |  |
| 学院(系) | 物联网学院 | 专 业 | 网络工程 |

**实 验 报 告**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **实验名称** | | **约束、视图、索引与存储过程** | | | | | **指导教师** | | |  |
| **实验类型** | | **验证** | | **实验学时** | | **4** | **实验时间** | | | 2024.12.9 |
| 1. **实验目的和要求**   (1) 通过上机实践，熟悉PostgreSQL的操作环境及使用方法。  (2) 掌握数据表的创建以及表的操作。  (3) 熟练掌握约束、视图、索引与存储过程的创建。 | | | | | | | | | | |
| 二、**实验环境(实验设备)**  硬件：微机  软件：Windows xp/Windows 7和PostgreSQL  素材：实验数据 | | | | | | | | | | |
| **三、实验内容**  1. 创建产品数据库products；  create database products  2. a)采用CREATE TABLE语句创建产品数据库products的关系模式：  **Product (maker, model, type)**  create table Product(  maker char(1),  model int primary key,  type varchar(10)  );  **PC (model, speed, ram, hd, price)**  create table PC(  model int primary key,  speed numeric(3,2),  ram int,  hd int,  price int  );  **Laptop (model, speed, ram, hd, screen, price)**  create table Laptop(  model int primary key,  speed numeric(3,2),  ram int,  hd int,  screen numeric(3,1),  price int  );  **Printer (model, color, type, price)**  create table Printer(  model int primary key,  color bool,  type varchar(10),  price int  );  b)采用COPY…FROM…语句将数据装入产品数据库；  copy laptop from 'C:\data\laptop.txt';  copy pc from 'C:\data\pc.txt';  copy printer from 'C:\data\printer.txt';  copy product from 'C:\data\product.txt';  3. 用alter table命令修改PC关系模式，增加参照完整性约束cpc，要求PC表中的model必须是在Product表中已有的记录。若违反了此约束，则拒绝更新操作。  alter table pc  add constraint cpc foreign key (model) references product(model);  4. 用alter table命令修改Laptop关系模式，增加完整性约束con\_screen，要求笔记本的屏幕大小不能小于13也不能大于22。  alter table laptop  add constraint con\_screen check (screen >= 13 and screen <= 22);  5. 用alter table命令修改Printer关系模式，增加完整性约束con\_type，要求type只能是laser或ink-jet中的一种。  alter table printer  add constraint con\_type check (type in ('laser', 'ink-jet'));  6. 用alter table命令修改PC关系模式，增加完整性约束con\_price，要求硬盘容量（hd）小于100的电脑价格不能超过600。  alter table pc  add constraint con\_price check (not (hd < 100 and price > 600));  7. 设计数据对3~6的内容进行验证。  **验证PC表中model的参照完整性约束cpc**  -- 测试数据  -- 插入一条在Product表中不存在的model数据  insert into pc (model, speed, ram, hd, price)  values (999, 3.5, 8, 256, 800);  -- 结果：插入失败，提示外键约束cpc被违反。    -- 修复数据  -- 先在Product表中添加对应的model数据  insert into product (maker, model, type)  values ('A', 999, 'pc');  -- 再插入到PC表  insert into pc (model, speed, ram, hd, price)  values (999, 3.5, 8, 256, 800);    **验证Laptop表中屏幕大小的完整性约束con\_screen**  -- 测试数据  -- 插入屏幕大小小于13的数据  insert into laptop (model, speed, ram, hd, screen, price)  values (1001, 2.4, 4, 128, 12.5, 700);  -- 插入屏幕大小大于22的数据  insert into laptop (model, speed, ram, hd, screen, price)  values (1002, 3.2, 16, 512, 23.0, 1200);  -- 结果：两次插入都失败，提示检查约束con\_screen被违反。      -- 修复数据  -- 插入符合条件的数据  insert into laptop (model, speed, ram, hd, screen, price)  values (1003, 2.8, 8, 256, 15.6, 900);    **验证Printer表中type类型的完整性约束con\_type**  -- 测试数据  -- 插入不符合类型的数据  insert into printer (model, color, type, price)  values (2001, true, 'dot-matrix', 300);  -- 插入dot-matrix的数据失败，提示检查约束con\_type被违反。    -- 插入符合类型的数据  insert into printer (model, color, type, price)  values (2002, false, 'laser', 500);  -- 插入laser的数据成功。    **验证PC表中硬盘容量和价格的完整性约束con\_price**  -- 测试数据  -- 插入硬盘容量小于100但价格超过600的数据  insert into pc (model, speed, ram, hd, price)  values (3001, 3.0, 4, 80, 700);  -- 插入失败，提示检查约束con\_price被违反。    -- 修改数据，使其价格符合约束  insert into pc (model, speed, ram, hd, price)  values (3003, 3.0, 4, 80, 600);    8. 用create view命令创建如下视图：  a) 视图Colorprinter给出所有彩色打印机的model、type和price；  create view colorprinter as  select model, type, price  from printer  where color = true;  b) 视图Ahighpc给出了A厂商生产的速度在2.0以上的PC的model和price。  create view ahighpc as  select pc.model, pc.price  from pc  join product on pc.model = product.model  where product.maker = 'A' and pc.speed > 2.0;  9. 用create index命令在Laptop的speed属性上创建索引spindex，顺序为降序。  create index spindex  on laptop (speed desc);  10. 编写存储过程(函数) ：  a) 在PC表中，将某一速度（整型）作为输入参数，求大于该速度的PC机的平均价格。  create or replace function meanprice(in pspeed int, out mean real)  as $$  declare  getprice int;  mcount int := 0;  total int := 0;  begin  mean := 0;  for getprice in select price from pc where speed > pspeed  loop  mcount := mcount + 1;  total := total + getprice;  end loop;  if mcount > 0 then  mean := total::real / mcount;  else  mean := null; -- 如果没有符合条件的数据，返回 null  end if;  end;  $$ language plpgsql;  b) 输入maker，统计该生产厂商生产的Laptop的最高速度和最低速度。  create or replace function laptop\_speed\_stats(in maker char(1), out max\_speed numeric, out min\_speed numeric)  as $$  begin  select max(speed), min(speed)  into max\_speed, min\_speed  from laptop  join product on laptop.model = product.model  where product.maker = maker;  if max\_speed is null or min\_speed is null then  raise notice 'No laptops found for maker %', maker;  end if;  end;  $$ language plpgsql;  调用存储过程(函数)，进行数据验证。   1. select meanprice(2);      1. select laptop\_speed\_stats('A');     **四、实验小结（写出实验过程中所遇到的问题和解决的办法，解决问题的过程中得到的经验和体会）**  本次实验中，我深入学习并实践了PostgreSQL数据库的基本操作，包括表的创建、数据的插入以及复杂约束条件的应用。在实验过程中，我遇到了一些问题并通过查阅资料和实际操作解决了这些问题，从中积累了宝贵的经验和体会。  首先，在创建数据表并添加约束时，由于对约束的定义和语法规则不够熟悉，我在添加外键约束和检查约束时多次出现语法错误。例如，在为PC表添加外键约束时，由于没有提前确保Product表中存在对应的model数据，导致插入数据失败。为了解决这个问题，我仔细检查了约束的定义并在执行插入操作之前，先确保Product表中包含相关的model数据。这让我认识到，数据的一致性和完整性是数据库操作中的关键，必须在定义约束之前全面了解数据表之间的关系。  其次，在验证约束条件的过程中，我发现插入的数据经常违反约束条件，例如屏幕尺寸超出范围或硬盘容量和价格的约束条件不符合要求。为了解决这些问题，我针对每个约束编写了详细的测试数据，并通过分析插入失败的提示信息逐步调整数据输入。此外，我还利用PostgreSQL的错误提示信息定位问题，并结合实验指导文档进行修改。这使我更加熟悉了约束的作用和实现机制，同时也加深了对逻辑约束语法的理解。  在创建视图和索引时，由于对视图的查询条件和索引的排序规则理解不足，我最初生成的视图结果和索引效果与预期不符。通过反复测试和调整查询语句，我逐步掌握了视图的创建方法以及索引对查询性能的优化作用。这一过程中，我认识到实验中的细心和耐心至关重要，尤其是在处理较为复杂的SQL语句时，必须一步步验证结果以确保操作的正确性。  最后，在编写存储过程和函数时，我对PL/pgSQL的语法不够熟悉，导致初始编写的存储过程无法正确执行。通过查阅官方文档和示例代码，我逐步完善了函数的逻辑结构，并成功实现了实验要求的功能。这一过程让我深刻体会到，熟练掌握数据库的脚本语言对实现复杂数据处理任务的重要性，同时也认识到良好的调试方法可以有效提高开发效率。  本次实验帮助我全面理解了PostgreSQL数据库的使用方法以及约束、视图、索引和存储过程的应用场景。在解决问题的过程中，我积累了丰富的经验，增强了对数据库系统实际操作的信心，同时也意识到进一步提升SQL编程能力和逻辑思维的重要性。这些收获将为我今后的学习和实际应用奠定坚实的基础。 | | | | | | | | | | |
| **五、指导教师评语** | | | | | | | | | | |
| **成 绩** |  | | **批阅人** | |  | | | **日 期** |  | |