

1实验报告

实验目的

实验环境描述

实验内容

实验步骤

(1) 使用 SQL 语句 创建 关系 数据库 表：

(2) 更改表 PERSON,

(3) 删除表 ROOM 的一个属性 Rarea。

(4) 取消表 PR。

(5) 为 ROOM 表创建按 R#降序排列的索引。

在index中出现对于索引，说明语句正常执行 (6) 为 PERSON 表创建按 P#升序排列的索引。

说明语句正常执行 (7) 创建表 PERSON 的按 Pname 升序排列的唯一性索引。

说明语句正常执行 (8) 取消 PERSON 表 P#升序索引。

1. 自我实践

a. 创建 数据库 表 CUSTOMERS(CID,CNAME,CITY,DISCNT),

b. 创建数据库表 ORDERS(ORDNA,MONTH,CID,AID,PID,QTY,DOLLARS)。

c. 增加数据库表 PRODUCTS 的三个属性列： CITY,QUANTITY,PRICE。

正常添加，说明语句正常执行

d. 为以上 4 个表建立各自的按主键增序排列的索引。

e. 取消步骤(4)建立的 4 个索引。

出现的问题

实验目的

本实验旨在熟悉并掌握使用SQL（结构化查询语言）进行数据库管理和操作的技能。通过本实验，能够学会：

1. 熟练使用SQL的数据定义语言（DDL）来创建和修改数据库中的基本表。
2. 掌握如何通过SQL语句增加、删除或修改表中的列。
3. 学习并实践创建表的索引以优化数据检索性能。
4. 理解并执行取消表、索引或表约束的操作。

实验环境描述

为了完成本实验，需要具备以下环境和工具：

1. **数据库管理系统（DBMS）**：PostgreSQL，这是一个开源的对象关系数据库系统，支持SQL标准。
2. **数据库管理工具**：pgAdmin，这是一个开源的、跨平台的数据库管理工具，用于管理PostgreSQL数据库。
3. **文本编辑器**：用于编写和保存SQL脚本。
4. **网络连接**：确保能够连接到PostgreSQL数据库服务器。

实验内容

本节实验的主要内容包括：

1. 使用 CREATE 语句创建基本表。
2. 更改基本表的定义，增加列，删除列，修改列的数据类型
3. 创建表的升降序索引。
4. 取消表、表的索引或表的约束。

实验步骤

(1) 使用 SQL 语句 创建 关系 数据库 表：

SQL:

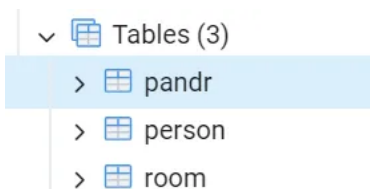
```
-- 创建人员表 PERSON
CREATE TABLE PERSON (
    P SERIAL PRIMARY KEY, -- P 为主键，并且是自动增长的序列号
    Pname VARCHAR(100) NOT NULL,
    Page INT CHECK (Page > 18), -- Page 必须大于 18
    Pgender VARCHAR(10) NOT NULL
);

-- 创建房间表 ROOM
CREATE TABLE ROOM (
    R SERIAL PRIMARY KEY, -- R 为主键，并且是自动增长的序列号
    Rname VARCHAR(100) NOT NULL,
    Rarea NUMERIC(10, 2) NOT NULL
);

-- 创建人员与房间关系表 P-R
CREATE TABLE PandR (
    P INT NOT NULL,
    R INT NOT NULL,
    Date DATE NOT NULL,
    PRIMARY KEY (P, R), -- 假设 P 和 R 联合作为主键
    FOREIGN KEY (P) REFERENCES PERSON(P) ON DELETE CASCADE, -- P 为外键，引用 PERSON 表的 P
    FOREIGN KEY (R) REFERENCES ROOM(R) ON DELETE CASCADE -- R 为外键，引用 ROOM 表的 R
);

-- 添加唯一性约束
ALTER TABLE PandR ADD CONSTRAINT unique_p_r UNIQUE (P, R);
```

结果:



在table项中出现新创建的三个表，说明语句正常执行

(2) 更改表 PERSON,

增加属性 Ptype(类型是 CHAR, 长度为 10), 取消 Page大于 18 的约束。把表 ROOM 中的属性 Rname 的数据类型改成长度为 30。

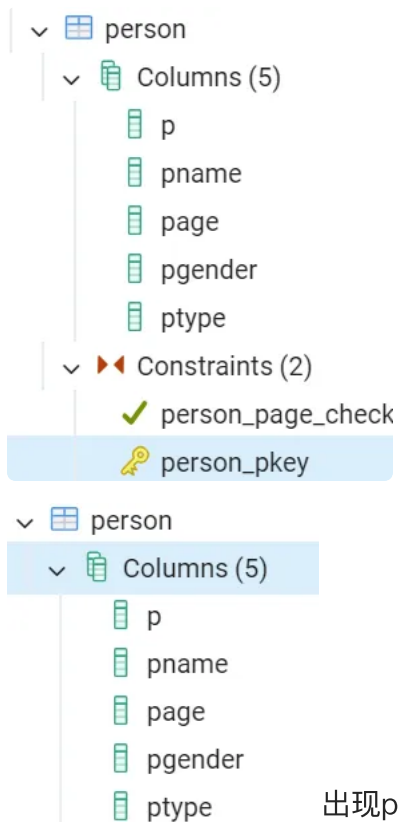
```

-- 为 PERSON 表添加新属性 Ptype
ALTER TABLE PERSON
ADD COLUMN Ptype CHAR(10);

-- 取消 PERSON 表中 Page 大于 18 的约束
ALTER TABLE PERSON
DROP CONSTRAINT IF EXISTS check_page;

-- 将 ROOM 表中 Rname 的数据类型改成长度为 30
ALTER TABLE ROOM
ALTER COLUMN Rname TYPE VARCHAR(30);|

```



出现ptype,并且增加限制, 说明语句正常执行

(3) 删除表 ROOM 的一个属性 Rarea。

2024/9/17 11:33:16	130 msec
Date	Rows affected
Duration	
Copy	Copy to Query Editor
<pre>ALTER TABLE ROOM DROP COLUMN Rarea;</pre>	

Messages

Query returned successfully in 130 msec.

(4) 取消表 PR。

Copy	Copy to Query Editor
<pre>DROP TABLE pandr CASCADE;</pre>	

Messages

Query returned successfully in 55 msec.

(5) 为 ROOM 表创建按 R#降序排列的索引。

```
CREATE INDEX idx_room_rnum_desc ON ROOM (R DESC);
```

Messages

Query returned successfully in 33 msec.

room

Columns

Constraints (1)

room_pkey

Indexes (1)

idx_room_rnum_desc

在index中出现对于索引，说明语句正常执行

(6) 为 PERSON 表创建按 P#升序排列的索引。

```
CREATE INDEX idx_person_pnum_asc ON PERSON (P ASC);
```

Messages

Query returned successfully in 44 msec.

说明语句正常执行

(7) 创建表 PERSON 的按 Pname 升序排列的唯一性索引。

```
CREATE UNIQUE INDEX idx_person_pname_asc ON PERSON (Pname ASC);
```

Messages

Query returned successfully in 34 msec.

说明语句正常

执行

(8) 取消 PERSON 表 P#升序索引。

```
DROP INDEX idx_person_pnum_asc
```

Messages

Query returned successfully in 37 msec.

说明语句正常执行

1. 自我实践

a. 创建数据库表 CUSTOMERS(CID,CNAME,CITY,DISCNT),

数据库表 AGENTS(AID,ANAME,CITY,PERCENT),数据库表 PRODUCTS(PID,PNAME)。其中, CID,AID,PID 分别是各表的主键, 具有唯一性约束。

```
-- 创建 CUSTOMERS 表
CREATE TABLE CUSTOMERS (
  CID INT PRIMARY KEY,      -- 客户ID, 主键
  CNAME VARCHAR(255),      -- 客户名称
  CITY VARCHAR(255),       -- 城市
  DISCNT DECIMAL(5, 2)     -- 折扣率
);
```

Messages

Query returned successfully in 34 msec.

```
-- 创建 AGENTS 表
CREATE TABLE AGENTS (
  AID INT PRIMARY KEY,     -- 代理商ID, 主键
  ANAME VARCHAR(255),      -- 代理商名称
  CITY VARCHAR(255),       -- 城市
  PERCENT DECIMAL(5, 2)   -- 百分比
);
```








Messages

Query returned successfully in 32 msec.

```
-- 创建 PRODUCTS 表
CREATE TABLE PRODUCTS (
  PID INT PRIMARY KEY,     -- 产品ID, 主键
  PNAME VARCHAR(255)      -- 产品名称
);
```

Messages

Query returned successfully in 37 msec.

- ▼  Tables (6)
 - >  agents
 - >  customers
 - >  orders
 - >  person
 - >  products
 - >  room

出现新建的几个表，说明语句正常执行

b.

创建数据库表 ORDERS(ORDNA,MONTH,CID,AID,PID,QTY,DOLLARS)。

其中，ORDNA 是主键，具有唯一性约束。CID,AID,PID 是外键，分别参照的是表 CUSTOMERS的 CID 字段，表 AGENTS 的 AID 字段，表 PRODUCTS 的 PID 字段。

```
CREATE TABLE ORDERS (  
  ORDNA INT PRIMARY KEY,  
  MONTH INT,  
  CID INT,  
  AID INT,  
  PID INT,  
  QTY INT,  
  DOLLARS DECIMAL(10, 2)  
);
```

Messages

Query returned successfully in 33 msec.

Tables (6)

- > agents
- > customers
- > orders
- > person
- > products
- > room



Plain Text

```
1 ALTER TABLE ORDERS  
2 ADD CONSTRAINT fk_orders_cid FOREIGN KEY (CID) REFERENCES CUSTOMERS(CID),  
3 ADD CONSTRAINT fk_orders_aid FOREIGN KEY (AID) REFERENCES AGENTS(AID),  
4 ADD CONSTRAINT fk_orders_pid FOREIGN KEY (PID) REFERENCES PRODUCTS(PID);
```

Messages

Query returned successfully in 30 msec.

说明语句正常执行

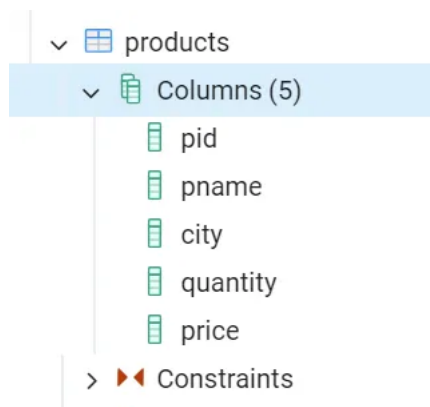
c.

增加数据库表 PRODUCTS 的三个属性列： CITY,QUANTITY,PRICE。


```
ALTER TABLE PRODUCTS
ADD COLUMN CITY VARCHAR(255),
ADD COLUMN QUANTITY INT,
ADD COLUMN PRICE DECIMAL(10, 2);
```

Messages

Query returned successfully in 36 msec.



正常添加，说明语句正常执行

d. 为以上 4 个表建立各自的按主键增序排列的索引。

```
-- 为 CUSTOMERS 表创建索引
CREATE INDEX idx_customers_cid ON CUSTOMERS (CID ASC);
```

Messages

Query returned successfully in 31 msec.

```

-- 为 AGENTS 表创建索引
CREATE INDEX idx_agents_aid ON AGENTS (AID ASC);

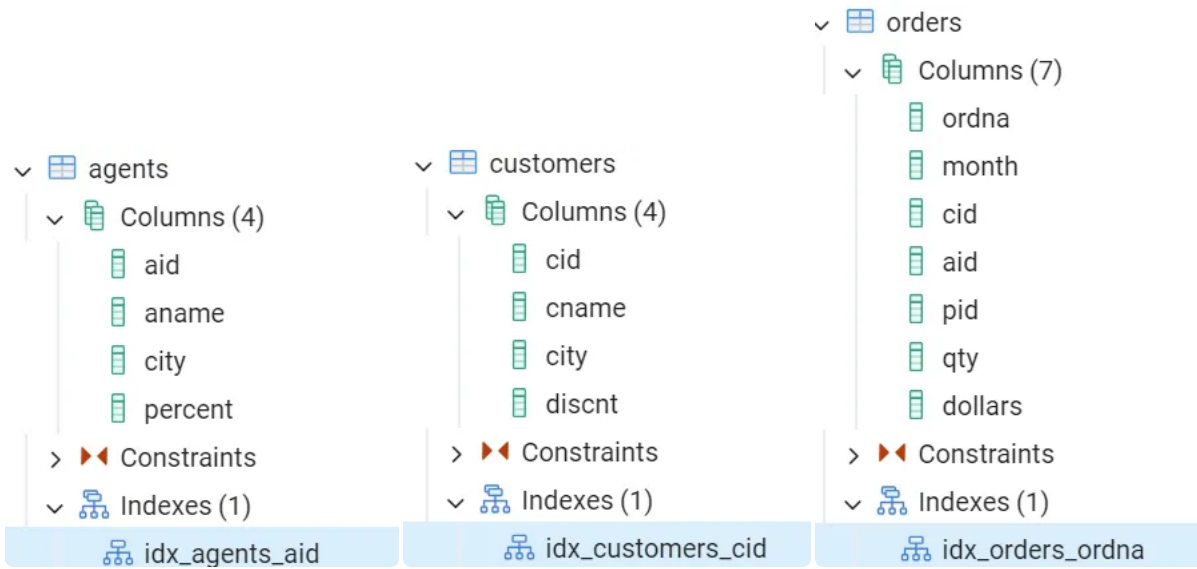
-- 为 PRODUCTS 表创建索引
CREATE INDEX idx_products_pid ON PRODUCTS (PID ASC);

-- 为 ORDERS 表创建索引
CREATE INDEX idx_orders_ordna ON ORDERS (ORDNA ASC);

```

Messages

Query returned successfully in 32 msec.



出现索引，说明语句正常执行

e. 取消步骤(4)建立的 4 个索引。

```
-- 取消 CUSTOMERS 表的索引
DROP INDEX IF EXISTS idx_customers_cid;

-- 取消 AGENTS 表的索引
DROP INDEX IF EXISTS idx_agents_aid;

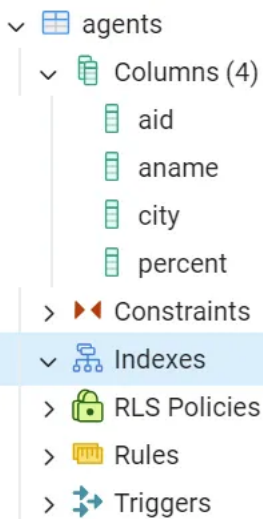
-- 取消 PRODUCTS 表的索引
DROP INDEX IF EXISTS idx_products_pid;

-- 取消 ORDERS 表的索引
DROP INDEX IF EXISTS idx_orders_ordna;
```

Messages

Query returned successfully in 32 msec.

if exist避免报错，先查询后删除



index中为空，说明语句正常执行

出现的问题

1. 注意标点切换成英文输入，中文括号会报错。