

实验二 数据定义

【姓名】谢宇桐

【学号】22336259

【专业】计算机科学与技术

1.实验目的

熟悉 SQL 的数据定义语言，能够熟练地使用 SQL 语句来创建和更改基本表，创建和取消索引。

2.实验环境

已安装完成Postgre SQL，在pgAdmin 4进行编辑

3.实验内容

本节实验的主要内容包括：

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

4.实验步骤（遇到的问题为灰色字）

(1)使用 SQL 语句创建关系数据库表：人员表 PERSON(P#,Pname,Page,Pgender)，房间表 ROOM(R#,Rname,Rarea)，表 P-R(P#,R#,Date)。其中 P#是表 PERSON 的主键，具有唯一性约束，Page 具有约束：大于 18; R#是表 ROOM 的主键，具有唯一性约束。表 P-R 中的 P#，R#是外键。

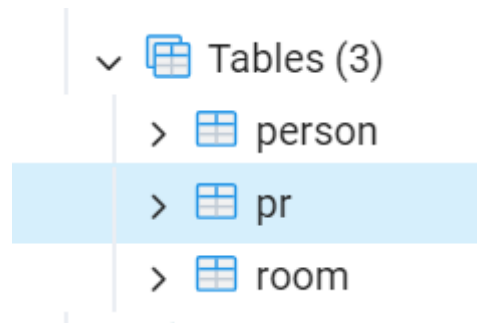
在这里P#无法作为主键出现在代码中，因此我将P#换为PER，R#换为R

```
CREATE TABLE PERSON(  
    PER char(20) NOT NULL UNIQUE,  
    Pname char(20) NOT NULL,  
    Page int,  
    Pgender char(10) ,  
    PRIMARY KEY(PER), CHECK(Page>18)  
);
```

```
CREATE TABLE ROOM(  
    R CHAR(8) NOT NULL UNIQUE,  
    Rname CHAR(20),  
    Rarea FLOAT(10),  
    PRIMARY KEY(R)  
);
```

```
CREATE TABLE PR(  
    PER char(20),
```

```
R char(20),  
Date timestamp without time zone,  
PRIMARY KEY (PER,R),  
FOREIGN KEY (PER) REFERENCES PERSON(PER),  
FOREIGN KEY (R) REFERENCES ROOM(R)  
);
```



(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。

```
1  --2  
2  ALTER TABLE PERSON ADD Ptype CHAR(10);  
3  ALTER TABLE PERSON DROP CONSTRAINT person_page_check;  
4  ALTER TABLE ROOM ALTER COLUMN Rname TYPE CHAR(30);  
5  
6
```

Data Output Messages Notifications

ALTER TABLE

Query returned successfully in 53 msec.

结果如下：

Data Output Messages Notifications

≡+ 📄 ▼ 📋 ▼ 🗑️ 🗄️ ⬇️ 📈 SQL

```
1 t WHERE conrelid = 'person'::regclass AND contype = 'c';
```

≡+ 📄 ▼ 📋 ▼ 🗑️ 🗄️ ⬇️ 📈 SQL

conname

name

通过下面代码查看rname长度

Query Query History

```
1 SELECT column_name, character_maximum_length
2 FROM information_schema.columns
3 WHERE table_name = 'room';
```

Data Output Messages Notifications

| | column_name name | character_maximum_length integer |
|---|---------------------|-------------------------------------|
| 1 | r | 8 |
| 2 | rname | 30 |

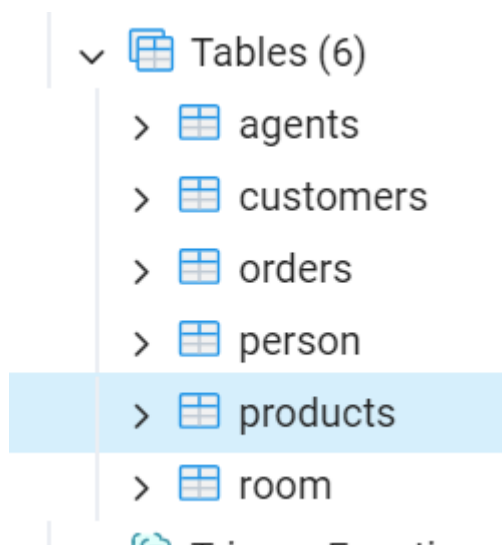
rname长度改为30（上图为完成第三操作后的表格，因此没有属性Rarea）

(3) 删除表 ROOM 的一个属性 Rarea，结果如上图。

```
--3
ALTER TABLE ROOM DROP COLUMN Rarea;
```

(4) 取消表 PR。

```
--4
DROP TABLE PR;
```



可以看到pr表格已经没了（多的表格是自我实践里的，这是最后检查时发现没有截图补的）

```
--5
CREATE INDEX XCNO ON ROOM(R DESC);
```

```
--6
CREATE INDEX XSNO ON PERSON(PER);
```

```
--7
CREATE UNIQUE INDEX RNUA ON PERSON(Pname ASC);
```

```
1 SELECT * FROM pg_indexes WHERE tablename = 'person';
```

Data Output Messages Notifications

≡+ 📄 ▼ 📋 ▼ 🗑️ 🗄️ ⬇️ 📈 SQL

| | schemaname name | tablename name | indexname name | tablespace name | indexdef text |
|---|--------------------|-------------------|-------------------|--------------------|--|
| 1 | public | person | person_pkey | [null] | CREATE UNIQUE INDEX person_pkey ON public.person USING btree (p... |
| 2 | public | person | xsno | [null] | CREATE INDEX xsno ON public.person USING btree (per) |
| 3 | public | person | rnua | [null] | CREATE UNIQUE INDEX rnua ON public.person USING btree (pname) |

```
--8
DROP INDEX public.XSNO;
```

1

SELECT * FROM pg_indexes WHERE tablename = 'person';

Data Output

Messages

Notifications

≡

📄

▼

📋

▼

🗑

🗄

⬇

📈

SQL

| | schemaname name | tablename name | indexname name | tablespace name | indexdef text |
|---|--------------------|-------------------|-------------------|--------------------|--|
| 1 | public | person | person_pkey | [null] | CREATE UNIQUE INDEX person_pkey ON pi |
| 2 | public | person | rnua | [null] | CREATE UNIQUE INDEX rnua ON public.per |

至此，课内实验成功完成。

5.“自我实践”实验步骤

(1) 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。

```
CREATE TABLE CUSTOMERS(
CID CHAR(20) NOT NULL UNIQUE,
CNAME CHAR(20),
CITY CHAR(20),
DISCNT INT,
PRIMARY KEY (CID)
);

CREATE TABLE AGENTS(
AID CHAR(20) NOT NULL UNIQUE,
ANAME CHAR(20),
CITY CHAR(20),
“PERCENT” FLOAT,
PRIMARY KEY (AID)
);

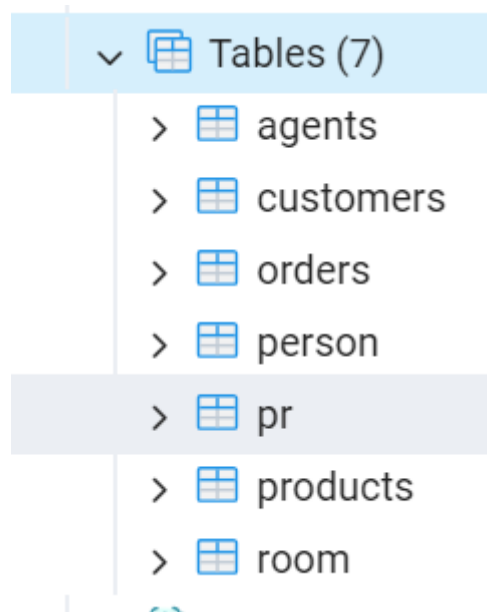
CREATE TABLE PRODUCTS(
PID CHAR(20) NOT NULL UNIQUE,
PNAME CHAR(20),
PRIMARY KEY (PID)
);
```

结果同 (2)

(2) 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 CHAR(20) NOT NULL UNIQUE,
“MONTH” INT,
CID CHAR(20),
AID CHAR(20),
PID CHAR(20),
```

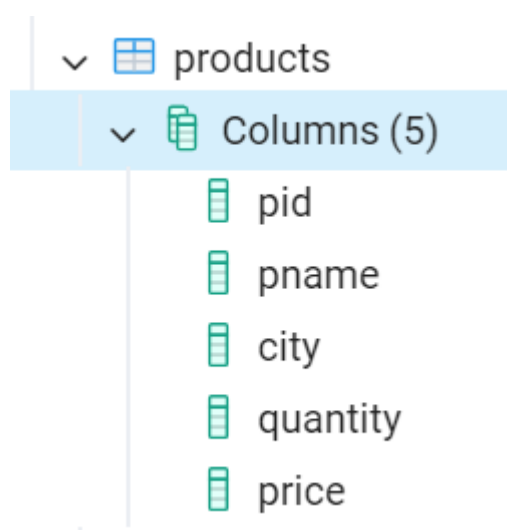
```
QTY INT,  
DOLLARS FLOAT,  
PRIMARY KEY (CID,AID,PID),  
FOREIGN KEY (CID) REFERENCES CUSTOMERS(CID),  
FOREIGN KEY (AID) REFERENCES AGENTS(AID),  
FOREIGN KEY (PID) REFERENCES PRODUCTS(PID)  
);
```



可以看到上述四个表格都创建成功

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

```
ALTER TABLE PRODUCTS ADD CITY CHAR(20);  
ALTER TABLE PRODUCTS ADD QUANTITY CHAR(20);  
ALTER TABLE PRODUCTS ADD PRICE CHAR(20);
```



可以看到创建三个属性列成功

(4) 为以上 4 个表建立各自的按主键增序排列的索引。

```
CREATE INDEX XCON ON CUSTOMERS(CID ASC);
CREATE INDEX XAON ON AGENTS(AID ASC);
CREATE INDEX XPON ON PRODUCTS(PID ASC);
CREATE INDEX XOON ON ORDERS(ORDNA ASC);
```

```
--使用这个代码查看是否创建成功
SELECT schemaname, tablename, indexname
FROM pg_indexes
WHERE tablename IN ('customers', 'agents', 'products', 'orders');
```

在这里当我将这几个表格名全部大写时会查不出来索引信息，换成小写则可以。据推测该信息是默认为小写保存，所以需要注意大小写区分。

| | schemaname name | tablename name | indexname name |
|---|--------------------|-------------------|-------------------|
| 1 | public | customers | customers_pkey |
| 2 | public | agents | agents_pkey |
| 3 | public | products | products_pkey |
| 4 | public | orders | orders_pkey |
| 5 | public | orders | orders_ordna_key |
| 6 | public | customers | xcon |
| 7 | public | agents | xaon |
| 8 | public | products | xpon |
| 9 | public | orders | xoon |

可以看到四个插入成功

(5) 取消步骤(4)建立的 4 个索引。

```
DROP INDEX public.XCON;
DROP INDEX public.XAON;
DROP INDEX public.XPON;
DROP INDEX public.XOON;
```



```
DROP INDEX customers.XCON;  
DROP INDEX agents.XAON;  
DROP INDEX products.XPON;  
DROP INDEX orders.XOON;
```

查看后可以看到我们成功删除这些索引：

Data Output Messages Notifications

| | schemaname name | tablename name | indexname name |
|---|---------------------------|--------------------------|--------------------------|
| 1 | public | customers | customers_pkey |
| 2 | public | agents | agents_pkey |
| 3 | public | products | products_pkey |
| 4 | public | orders | orders_pkey |
| 5 | public | orders | orders_ordna_key |