

实验二

实验目的

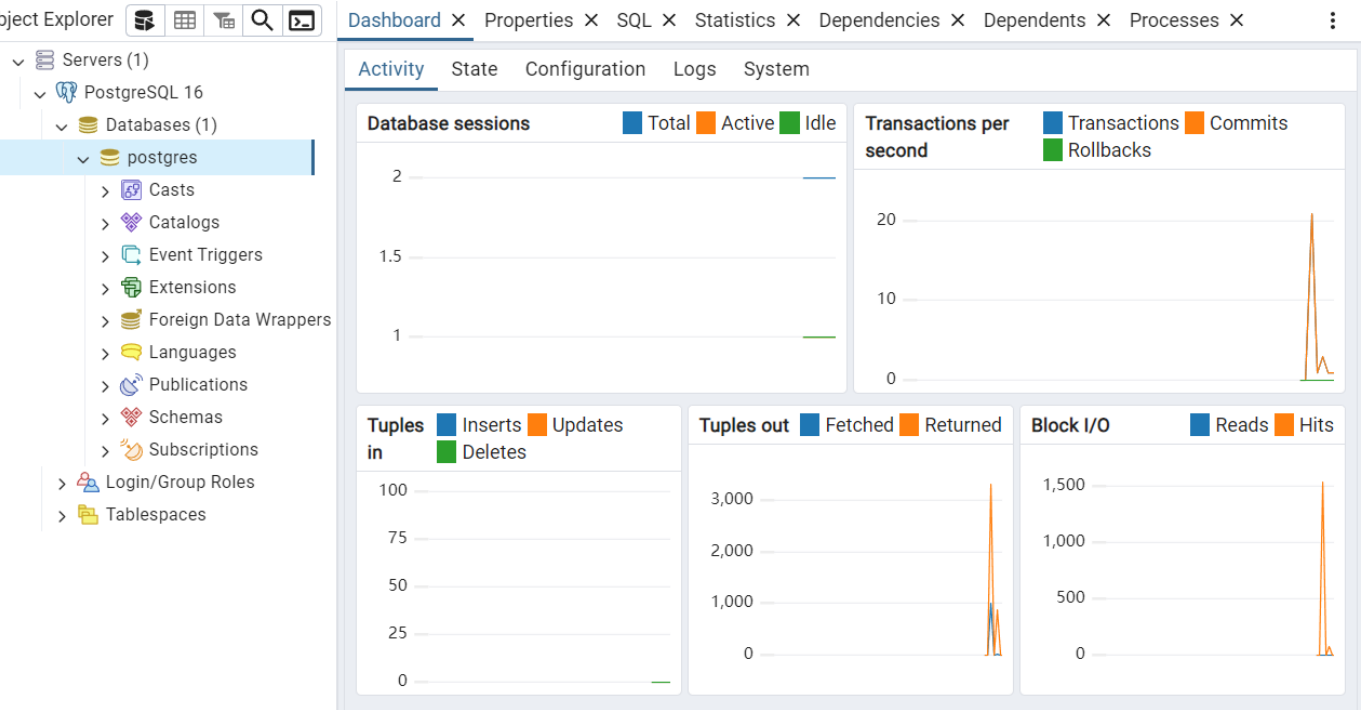
熟悉SQL的数据定义语言，能够熟练地使用SQL语句来创建和更改基本表，创建和取消索引。

实验环境

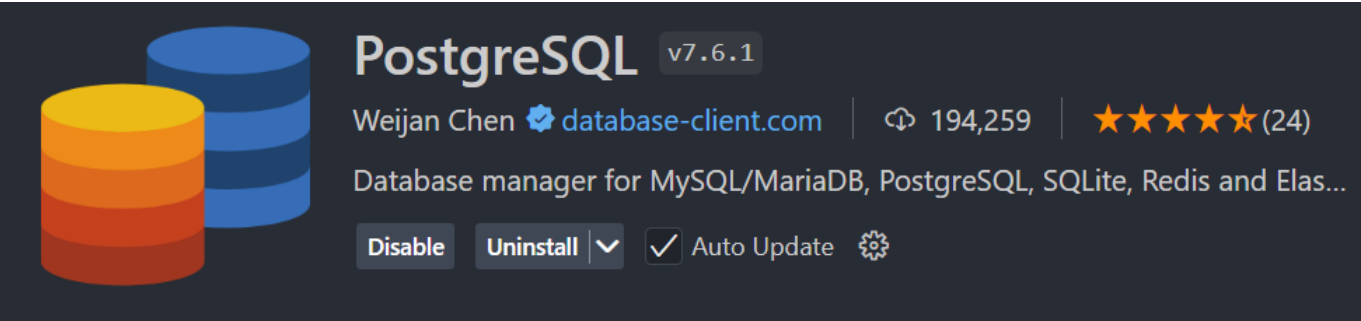
- OS: Windows 11

```
OsName           : Microsoft Windows 11 企业版
OsType           : WINNT
OsOperatingSystemSKU : EnterpriseEdition
OsVersion        : 10.0.22631
```

- Database: PostgreSQL 16



- IDE: Visual Studio Code (with plugin PostgreSQL)



实验内容

- 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。

实验步骤

1. 使用SQL语句创建关系数据库表。

- 人员关系表 `PERSON(P#, Pname, Page, Pgender)`，其中P#为主键，Page具有约束：大于18

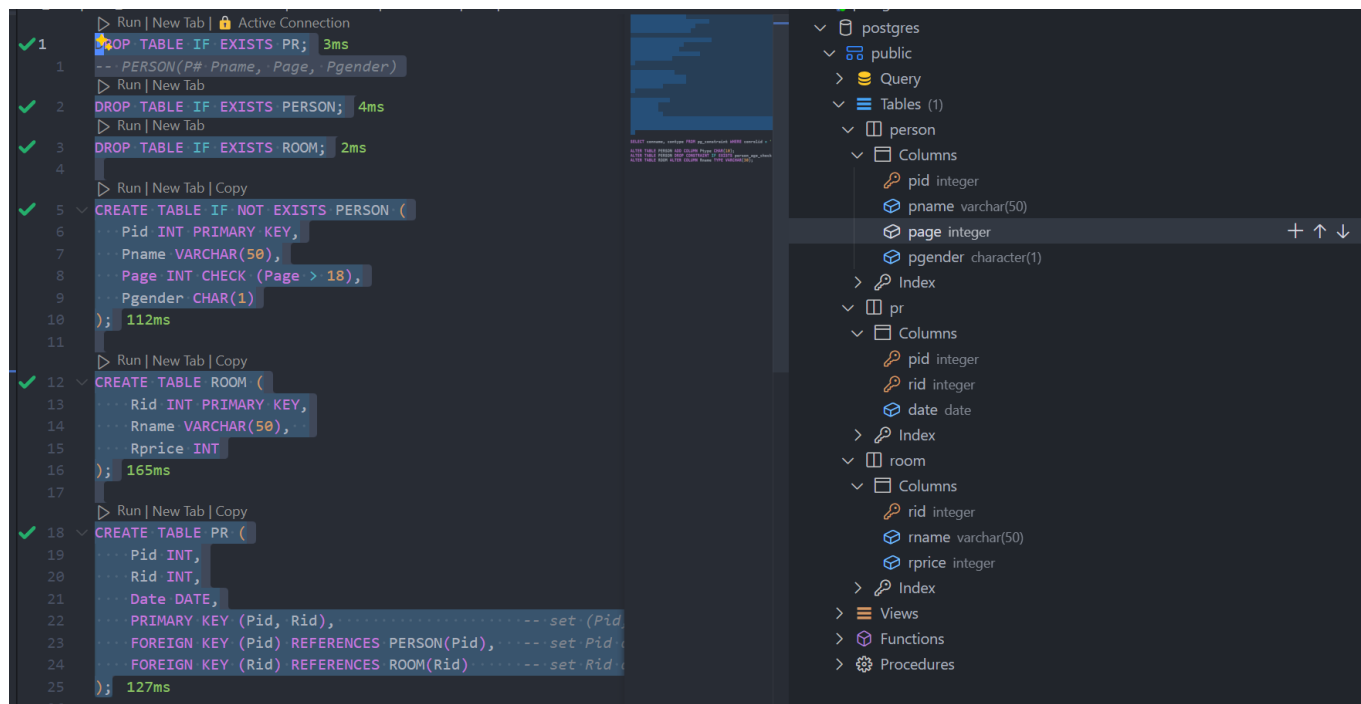
```
CREATE TABLE IF NOT EXISTS PERSON (  
    Pid INT PRIMARY KEY,  
    Pname VARCHAR(50),  
    Page INT CHECK (Page > 18),  
    Pgender CHAR(1)  
);
```

- 房间表 `ROOM(R#, Rname, Rarea)`，其中R#为主键

```
CREATE TABLE IF NOT EXISTS ROOM (  
    Rid INT PRIMARY KEY,  
    Rname VARCHAR(50),  
    Rarea INT  
);
```

- 关系表P-R `PR(P#, R#, Date)`，其中P#和R#为外键

```
CREATE TABLE IF NOT EXISTS PR (  
    Pid INT,  
    Rid INT,  
    Date DATE,  
    PRIMARY KEY (Pid, Rid),           -- set (Pid, Rid) as primary key  
    FOREIGN KEY (Pid) REFERENCES PERSON(Pid), -- set Pid as foreign key to  
PERSON table  
    FOREIGN KEY (Rid) REFERENCES ROOM(Rid)    -- set Rid as foreign key to ROOM  
table  
);
```



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。

```

ALTER TABLE PERSON ADD COLUMN Ptype CHAR(10);
-- When setting the constraint condition for Page, I did not set a name for it.
-- However, it is not a problem, we can easily get the name of the anonymous
constraint
-- by running the following statement:
-- `SELECT conname FROM pg_constraint WHERE conrelid = 'PERSON'::regclass AND
contype = 'c';
-- Then we can get the name of the constraint that the database assigns to us by
default,
-- which is `person_page_check`.
ALTER TABLE PERSON ALTER COLUMN Page DROP CONSTRAINT IF EXISTS person_page_check;
ALTER TABLE ROOM ALTER COLUMN Rname TYPE VARCHAR(30);

```

Run | New Tab | JSON

28 SELECT conname, contype FROM pg_constraint WHERE conrelid = 'PERSON'::regclass AND contype = 'c'; 3ms

1 Run | New Tab

2 ALTER TABLE PERSON ADD COLUMN Ptype CHAR(10);

Run | New Tab

3 ALTER TABLE PERSON ALTER COLUMN Page DROP CONSTRAINT IF EXISTS person_page_check;

Run | New Tab

4 ALTER TABLE ROOM ALTER COLUMN Rname TYPE VARCHAR(30);

pg_catalog.pg_constraint X

Search results

Cost: 5ms < 1 > Total 1

	* conname name	* contype "char"
	Filter	Filter
	> person_page_check	c

Database > experiment > expr2 > expr2.sql > ALTER TABLE PERSON ADD COLUMN Ptype CHAR(10)

Run | New Tab | Copy

17 CREATE TABLE ROOM (

16 Rid INT PRIMARY KEY,

15 Rname VARCHAR(50),

14 Rprice INT

13);

12

Run | New Tab | Copy

11 CREATE TABLE PR (

10 Pid INT,

9 Rid INT,

8 Date DATE,

7 PRIMARY KEY (Pid, Rid),

6 FOREIGN KEY (Pid) REFERENCES PERSON(Pid),

5 FOREIGN KEY (Rid) REFERENCES ROOM(Rid)

4);

3

Run | New Tab | JSON

2 SELECT conname, contype FROM pg_constraint WHERE conrelid =

1

Run | New Tab

30 ALTER TABLE PERSON ADD COLUMN Ptype CHAR(10); 1ms

Run | New Tab

1 ALTER TABLE PERSON DROP CONSTRAINT IF EXISTS person_age_che

Run | New Tab

2 ALTER TABLE ROOM ALTER COLUMN Rname TYPE VARCHAR(30); 42ms

postgres 16.4

postgres

public

Query

Tables (1)

person

Columns

pid integer

pname varchar(50)

page integer

pgender character(1)

ptype character(10)

Index

pr

Columns

pid integer

rid integer

date date

Index

room

Columns

rid integer

rname varchar(30)

rprice integer

Index

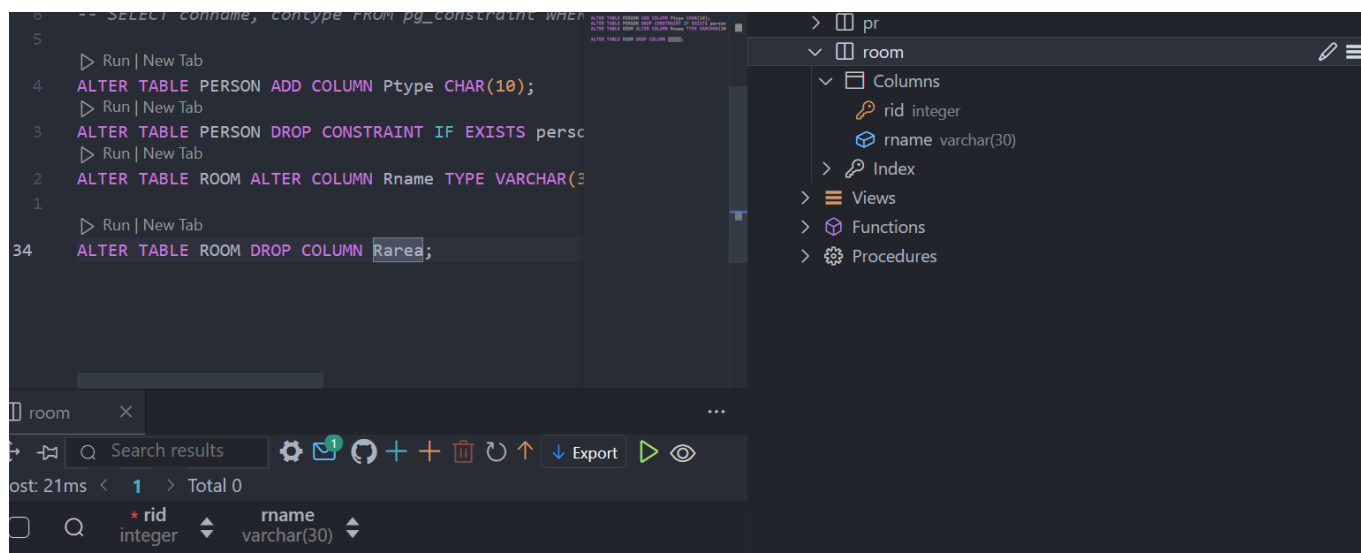
Views

Functions

Procedures

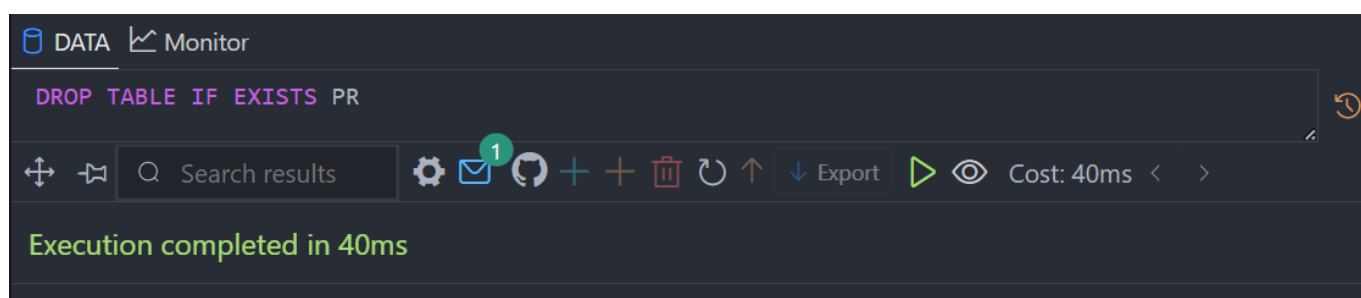
3. 删除表ROOM中的一个属性Rarea。

ALTER TABLE ROOM DROP COLUMN Rarea;



4. 取消表PR。

```
DROP TABLE IF EXISTS PR;
```

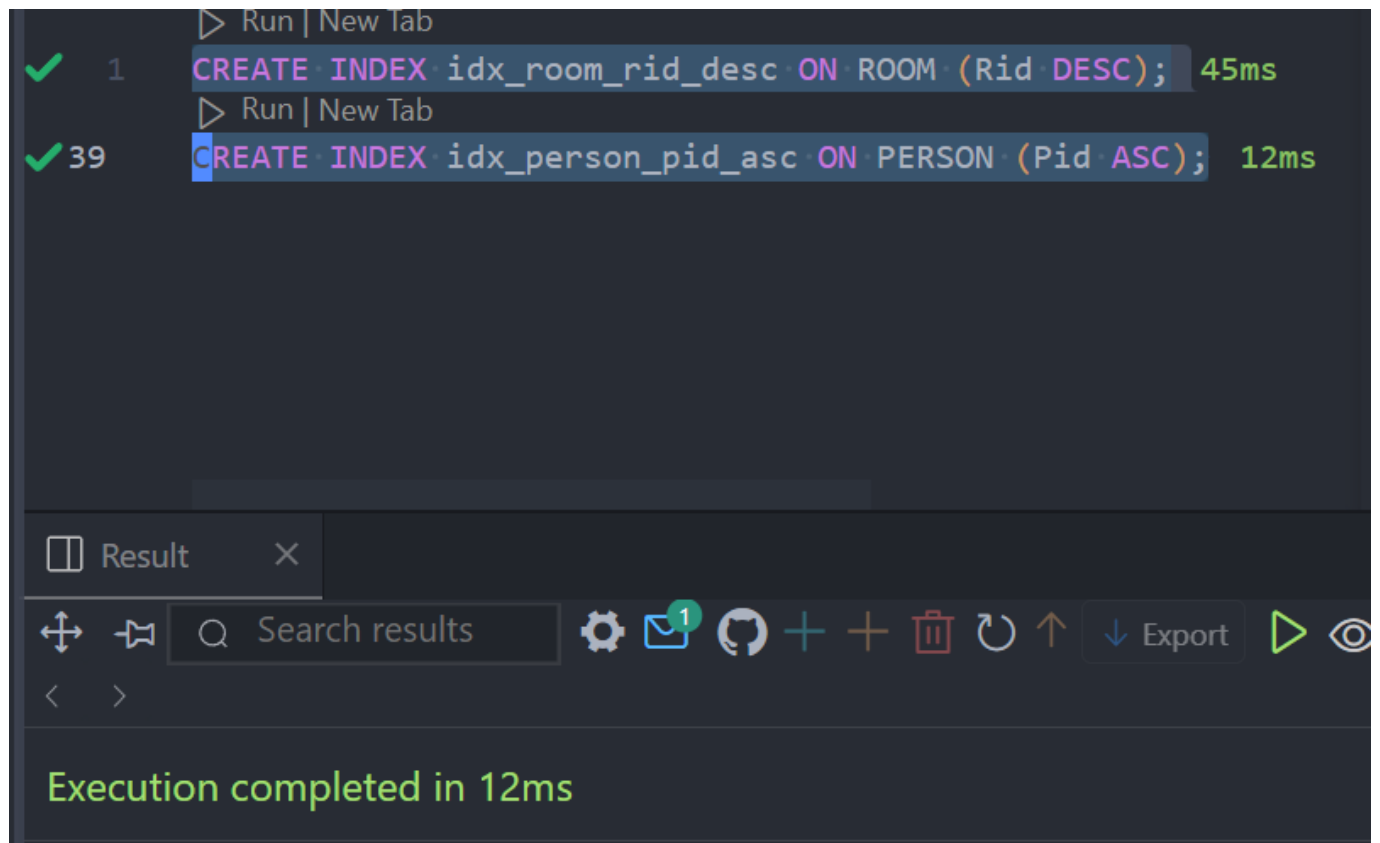


5. 为ROOM表创建按R#降序的索引。

```
CREATE INDEX idx_room_rid_desc ON ROOM (Rid DESC);
```

6. 为PERSON表创建按P#升序的索引。

```
CREATE INDEX idx_person_pid_asc ON PERSON (Pid ASC);
```



```
Run | New Tab
✓ 1 CREATE INDEX idx_room_rid_desc ON ROOM (Rid DESC); 45ms
Run | New Tab
✓ 39 CREATE INDEX idx_person_pid_asc ON PERSON (Pid ASC); 12ms
```

Result X

Search results

Execution completed in 12ms

7. 创建表PERSON的按Pname升序排序的唯一性索引。

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

8. 取消PERSON表P#的升序索引。

```
DROP INDEX idx_person_pid_asc;
```

The screenshot displays a SQL script with 6 steps, executed in a development environment. The script is as follows:

```

4. 取消表`PR`。

```sql
DROP TABLE IF EXISTS PR;
```

![[007]](/img/007.png)

5. 为`ROOM`表创建按`R#`降序的索引。

```sql
CREATE INDEX idx_room_rid_desc ON ROOM (Rid DESC);
```

6. 为`PERSON`表创建按`P#`升序的索引。

```sql
CREATE INDEX idx_person_pid_asc ON PERSON (Pid ASC);
```

```

The execution results show the following steps and their completion times:

- Step 1: `CREATE INDEX idx_room_rid_desc ON ROOM (Rid DESC);` 269ms
- Step 2: `CREATE INDEX idx_person_pid_asc ON PERSON (Pid ASC);` 213ms
- Step 3: `CREATE UNIQUE INDEX idx_pr_date_asc ON PR (Date ASC);` 218ms
- Step 4: `DROP INDEX idx_person_pid_asc;` 96ms

The final output shows "Execution completed in 96ms".

自我实践

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。
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。
3. 增加数据库表PRODUCTS的三个属性列: CITY, QUANTITY, PRICE。
4. 为以上4个表建立各自的按主键增序排列的索引。
5. 取消步骤4建立的4个索引。

```
DROP TABLE IF EXISTS ORDERS;  
DROP TABLE IF EXISTS PRODUCTS;  
DROP TABLE IF EXISTS CUSTOMERS;  
DROP TABLE IF EXISTS AGENTS;
```

```
CREATE TABLE CUSTOMERS (
  CID      INT PRIMARY KEY,
  CNAME    VARCHAR(50),
  CITY     VARCHAR(50),
  DISCNT   DECIMAL(5,2))
```

```
);

CREATE TABLE AGENTS (
  AID      INT PRIMARY KEY,
  ANAME    VARCHAR(50),
  CITY     VARCHAR(50),
  PERCENT  DECIMAL(5,2)
);

CREATE TABLE PRODUCTS (
  PID      INT PRIMARY KEY,
  PNAME    VARCHAR(50)
);

CREATE TABLE ORDERS (
  ORDNA    INT PRIMARY KEY,
  MONTH    VARCHAR(50),
  CID      INT,
  AID      INT,
  PID      INT,
  QTY      INT,
  DOLLARS  DECIMAL(10,2),
  FOREIGN KEY (CID) REFERENCES CUSTOMERS(CID),
  FOREIGN KEY (AID) REFERENCES AGENTS(AID),
  FOREIGN KEY (PID) REFERENCES PRODUCTS(PID)
);

ALTER TABLE PRODUCTS ADD COLUMN CITY VARCHAR(50);
ALTER TABLE PRODUCTS ADD COLUMN QUANTITY INT;
ALTER TABLE PRODUCTS ADD COLUMN PRICE DECIMAL(10,2);

CREATE INDEX idx_products_pid_asc ON PRODUCTS (PID ASC);
CREATE INDEX idx_customers_cid_asc ON CUSTOMERS (CID ASC);
CREATE INDEX idx_agents_aid_asc ON AGENTS (AID ASC);
CREATE INDEX idx_orders_ordna_asc ON ORDERS (ORDNA ASC);

DROP INDEX idx_products_pid_asc;
DROP INDEX idx_customers_cid_asc;
DROP INDEX idx_agents_aid_asc;
DROP INDEX idx_orders_ordna_asc;
```


5

CNAME VARCHAR(50),

6

CITY VARCHAR(50),

7

DISCNT DECIMAL(5,2)

8

); 111ms

9

1

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Run | New Tab | Copy

19

CREATE TABLE AGENTS (

2

AID INT PRIMARY KEY,

3

ANAME VARCHAR(50),

4

CITY VARCHAR(50),

5

PERCENT DECIMAL(5,2)

6

); 356ms

7

Run | New Tab

7

CREATE TABLE PRODUCTS (

8

PID INT PRIMARY KEY,

9

PNAME VARCHAR(50)

10

); 347ms

11

Run | New Tab | Copy

12

CREATE TABLE ORDERS (

13

ORDNA INT PRIMARY KEY,

14

MONTH VARCHAR(50),

15

CID INT,

16

AID INT,

17

PID INT,

18

QTY INT,

19

DOLLARS DECIMAL(10,2),

20

FOREIGN KEY (CID) REFERENCES CUSTOMERS(CID),

21

FOREIGN KEY (AID) REFERENCES AGENTS(AID),

22

FOREIGN KEY (PID) REFERENCES PRODUCTS(PID)

23

); 337ms

24

Result

Cost: 9ms

Execution completed in 9ms

2024-09-15 22:50:17 [INFO] Executing: DROP TABLE IF EXISTS ORDERS

2024-09-15 22:50:17 [INFO] Result: Empty set in 4ms

2024-09-15 22:50:17 [INFO] Executing: DROP TABLE IF EXISTS PRODUCTS

2024-09-15 22:50:17 [INFO] Result: Empty set in 9ms

2024-09-15 22:50:17 [INFO] Executing: DROP TABLE IF EXISTS CUSTOMERS

2024-09-15 22:50:17 [INFO] Result: Empty set in 23ms

2024-09-15 22:50:17 [INFO] Executing: DROP TABLE IF EXISTS AGENTS

2024-09-15 22:50:17 [INFO] Result: Empty set in 9ms

2024-09-15 22:50:17 [INFO] Executing: CREATE TABLE CUSTOMERS ((CID INT PRIMARY KEY,

2024-09-15 22:50:17 [INFO] Result: Empty set in 111ms

2024-09-15 22:50:17 [INFO] Executing: CREATE TABLE AGENTS ((AID INT PRIMARY KEY, ANAME

2024-09-15 22:50:17 [INFO] Result: Empty set in 356ms

2024-09-15 22:50:17 [INFO] Executing: CREATE TABLE PRODUCTS ((PID INT PRIMARY KEY,

2024-09-15 22:50:18 [INFO] Result: Empty set in 347ms

2024-09-15 22:50:18 [INFO] Executing: CREATE TABLE ORDERS ((ORDNA INT PRIMARY KEY, MONTH

2024-09-15 22:50:18 [INFO] Result: Empty set in 337ms

2024-09-15 22:50:18 [INFO] Executing: ALTER TABLE PRODUCTS ADD COLUMN CITY VARCHAR(50)

2024-09-15 22:50:18 [INFO] Result: Empty set in 1ms

2024-09-15 22:50:18 [INFO] Executing: ALTER TABLE PRODUCTS ADD COLUMN QUANTITY INT

2024-09-15 22:50:18 [INFO] Result: Empty set in 44ms

2024-09-15 22:50:18 [INFO] Executing: ALTER TABLE PRODUCTS ADD COLUMN PRICE DECIMAL(10,2)

2024-09-15 22:50:18 [INFO] Result: Empty set in 2ms

2024-09-15 22:50:18 [INFO] Executing: CREATE INDEX idx_products_pid_asc ON PRODUCTS (PID ASC)

2024-09-15 22:50:19 [INFO] Result: Empty set in 285ms

2024-09-15 22:50:19 [INFO] Executing: CREATE INDEX idx_customers_cid_asc ON CUSTOMERS (CID ASC)

2024-09-15 22:50:19 [INFO] Result: Empty set in 347ms

2024-09-15 22:50:19 [INFO] Executing: CREATE INDEX idx_agents_aid_asc ON AGENTS (AID ASC)

2024-09-15 22:50:19 [INFO] Result: Empty set in 247ms

2024-09-15 22:50:19 [INFO] Executing: CREATE INDEX idx_orders_ordna_asc ON ORDERS (ORDNA ASC)

2024-09-15 22:50:19 [INFO] Result: Empty set in 243ms

2024-09-15 22:50:19 [INFO] Executing: DROP INDEX idx_products_pid_asc

2024-09-15 22:50:20 [INFO] Result: Empty set in 15ms

2024-09-15 22:50:20 [INFO] Executing: DROP INDEX idx_customers_cid_asc

2024-09-15 22:50:20 [INFO] Result: Empty set in 106ms

2024-09-15 22:50:20 [INFO] Executing: DROP INDEX idx_agents_aid_asc

2024-09-15 22:50:20 [INFO] Result: Empty set in 4ms

2024-09-15 22:50:20 [INFO] Executing: DROP INDEX idx_orders_ordna_asc