实验 基于 SQL 的数据定义与修改

1. 实验目的

(1) 掌握用 SQL 语句创建数据库、修改数据库属性、删除数据库的方法。
(2) 掌握用 SQL 语句对数据进行增、删、改等操作。
(3) 致敬英雄楷模，立志报效祖国。

2. 实验环境（写清硬件配置和软件版本）

(1) 硬件：戴尔灵越 7370，内存 16G，处理器 Intel i7 低功耗版，硬盘 512G 固态
(2) 操作系统：Win10 64 位
(3) 数据库管理系统及图形化管理工具：PostgreSQL 10，pgAdmin4

3. 实验内容

(1) 使用 SQL 语句创建数据库、修改数据库属性、删除数据库。
(2) 使用 SQL 语句插入数据、更新数据、删除数据。

4. 实验数据

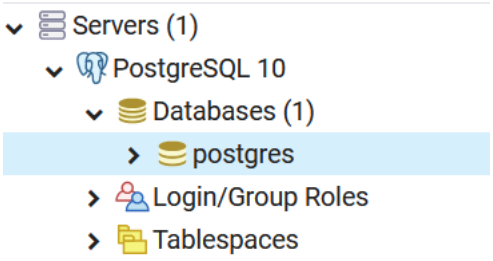
2019 年 12 月以来，湖北省武汉市部分医院陆续发现了多例有华南海鲜市场暴露史的不明原因肺炎病例，现已证实为2019新型冠状病毒感染引起的急性呼吸道传染病。当地时间 2020 年 3 月 11 日，世界卫生组织总干事谭德塞宣布，根据评估，世卫组织认为当前新冠肺炎疫情可被称为全球大流行（pandemic）。美国约翰斯·霍普金斯大学：截至北京时间 2020 年 9 月 23 日 7 时 23 分，全球新冠确诊病例达 31453048 例，死亡病例为 967347 例；美国是全球疫情最严重的国家，确诊病例达 6890014 例，死亡病例为 200654 例。新型冠状病毒肺炎爆发，面对疫病，英雄的中国人民非但没有被吓倒，反而众志成城、守望相助，树立起必胜的信心，在党和政府的坚强领导下，依赖科学、组织抗击，在短时间内取得抗疫斗争的胜利。学习抗疫英雄，争当时代先锋。本次实验基于“抗疫英雄数据库”，其中，抗疫英雄表包含英雄编号（aehero_id），名字（aehero_name），性别（aehero_gender）；抗疫措施表包含措施编号（aemeasure_id），措施名称(aemeasure_name)，措施具体内容（aemeasure_detail）；一位抗疫英雄会参与多项抗疫措施的制定，一项抗疫措施会有多位英雄参与制定，措施制定贡献表记录各位英雄参与各项措施制定中的贡献（aehero_deeds）。

5. 实验作业

(1) 在数据库 Antiepidemic 中，使用 SQL 语句创建数据表 aehero(aehero_id, aehero_name, aehero_gender)，设置 aehero_id 为主键。

① 创建数据库 Antiepidemic

开始时没有 Antiepidemic 数据库。

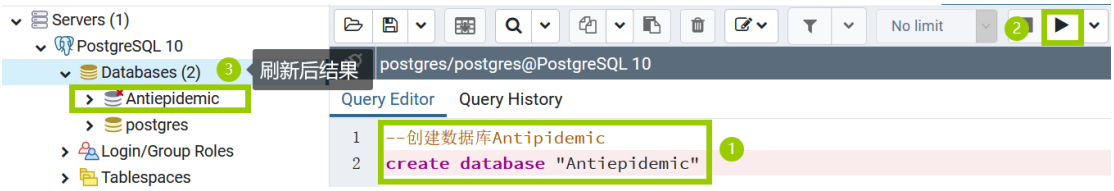


打开 Query Tools，输入注释和源代码，创建 Antiepidemic 数据库，单击运行，刷新 Databases 发现新创建的库，源代码、执行步骤与执行结果如图所示。

本题源代码：

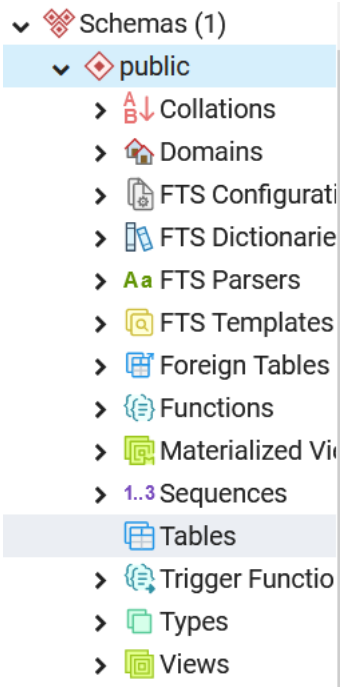
--创建数据库 Antipidemic

create database "Antiepidemic"



② 使用 SQL 语句创建数据表 aehero

操作前 Antiepidemic 数据库中没有该表。



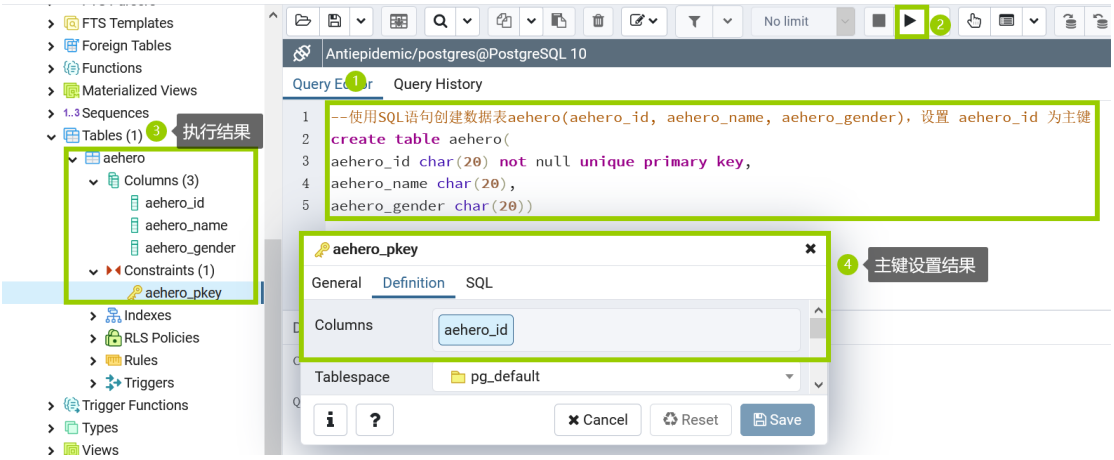
打开 Antiepidemic 数据库的 Query Tool，根据建表要求输入代码，其中将 aehero_id 设为主键，点击运行，刷新后查看结果，数据库下出现 aehero 数据表，Columns 正确，主键约束正确。

本题源代码：

--使用 SQL 语句创建数据表 aehero(aehero_id, aehero_name, aehero_gender)，设置 aehero_id 为主键

create table aehero(

aehero_id char(20) not null unique primary key,
aehero_name char(20),
aehero_gender char(20))



(2) 在数据库 Antiepidemic 中,使用 SQL 语句创建数据表 aemeasure(aemeasure_id, aemeasure_name, aemeasure_detail)，设置 aemeasure_id 为主键。

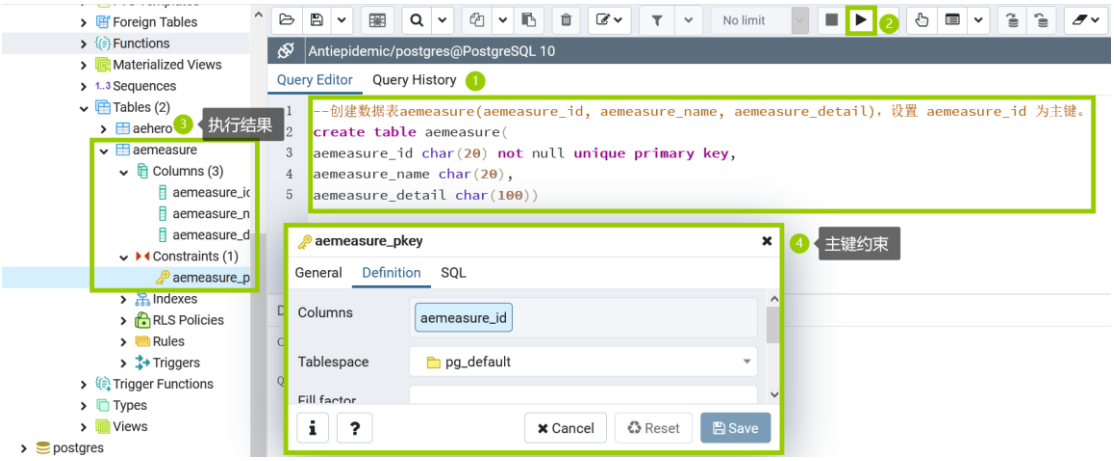
操作前数据库 Antiepidemic 中数据表状态如上题截图所示。

打开 Antiepidemic 数据库的 Query Tool，根据建表要求输入代码，其中将 aemeasure_id 设为主键，点击运行，刷新后查看结果，数据库下出现 aemeasure 数据表，Columns 正确，主键约束正确。

本题源代码：

--创建数据表 aemeasure(aemeasure_id, aemeasure_name, aemeasure_detail)，设置 aemeasure_id 为主键。

create table aemeasure(
aemeasure_id char(20) not null unique primary key,
aemeasure_name char(20),
aemeasure_detail char(100))



(3) 在数据库 Antiepidemic 中,使用 SQL 语句创建数据表 contribution(aehero_id, aemeasure_id, aehero_deeds)，设置主键、外键。

数据表 contribution 记录的是抗疫英雄在抗议措施中的贡献，而一位抗疫英雄会参与

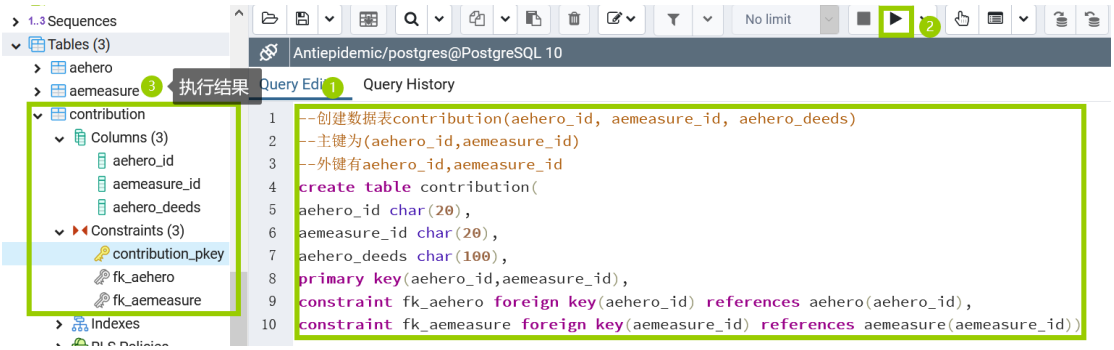
多项抗疫措施的制定、一项抗疫措施会有多位英雄参与制定，因此 aehero_id 和 aemeasure_id 共同作为主键。

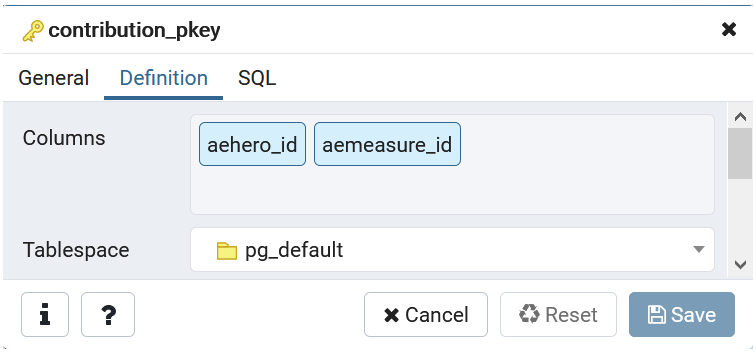
考虑外键设置，aehero_id 依赖 aehero.aehero_id，aemeasure_id 依赖 aemeasure_id，而 aehero_id 和 aemeasure_id 分别是所在数据表的主键、符合外键要求，因此该数据表有两个外键。

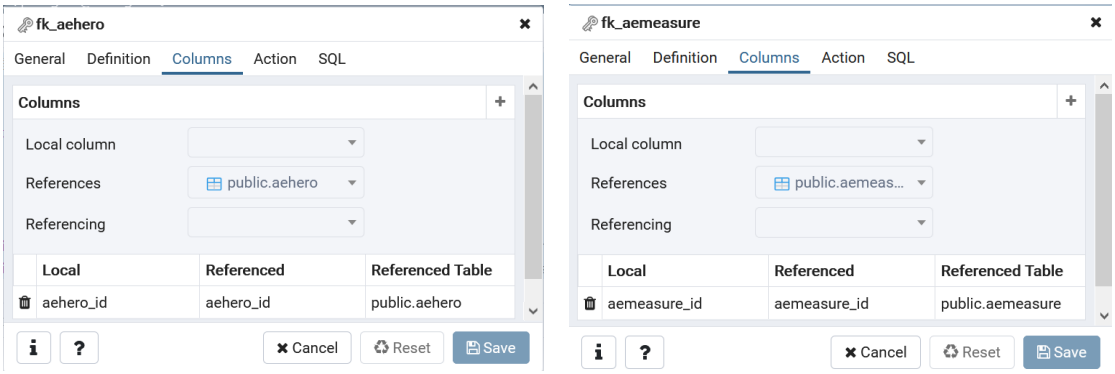
打开 Antiepidemic 数据库的 Query Tool，根据建表要求输入代码，点击运行，刷新后查看结果，数据库下出现 contribution 数据表，Columns 正确，主键、外键约束正确，源代码截图、运行结果截图、主键外键约束截图如下所示。

本题源代码：

```
--创建数据表 contribution(aehero_id, aemeasure_id, aehero_deeds)
--主键为(aehero_id,aemeasure_id)
--外键有 aehero_id,aemeasure_id
create table contribution(
aehero_id char(20),
aemeasure_id char(20),
aehero_deeds char(100),
primary key(aehero_id,aemeasure_id),
constraint fk_aehero foreign key(aehero_id) references aehero(aehero_id),
constraint fk_aemeasure foreign key(aemeasure_id) references aemeasure(aemeasure_id))
```

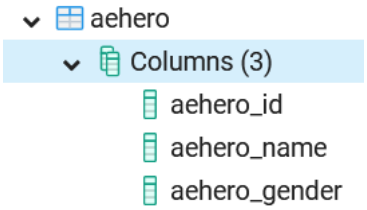






(4) 使用 SQL 语句修改数据表 aehero 的名称为 hero1，并添加字段 aehero_team（抗疫英雄所属医疗队）。

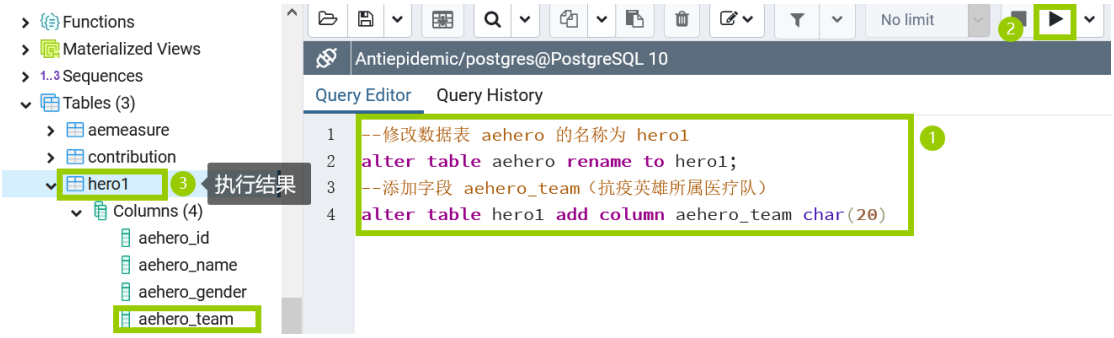
操作前数据表 aehero 的名称和字段如图所示。



打开数据库 Antiepidemic 的 Query Tool，根据题目要求输入 SQL 语句代码，单击运行，刷新后查看执行结果，数据表名称修改正确，字段成功添加。

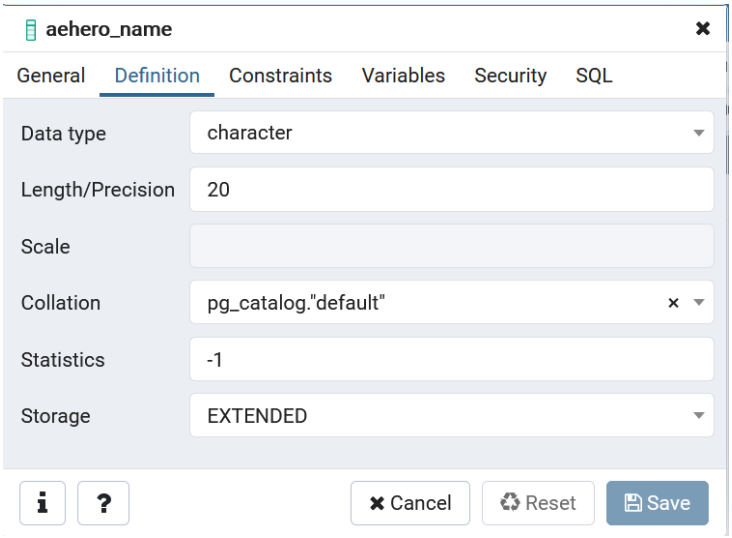
源代码如下：

```
--修改数据表 aehero 的名称为 hero1
alter table aehero rename to hero1;
--添加字段 aehero_team（抗疫英雄所属医疗队）
alter table hero1 add column aehero_team char(20)
```



(5) 使用 SQL 语句修改数据表 hero1 中的 aehero_name 字段的类型为文本类型。

修改前 aehero_name 的类型为 char(20)

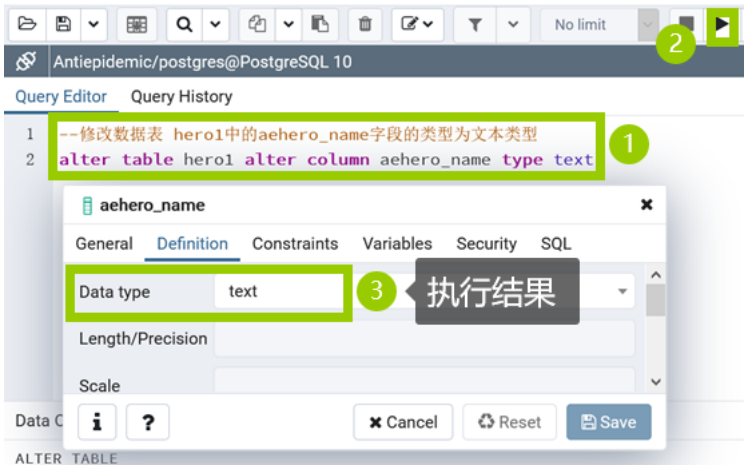


打开数据库 Antiepidemic 的 Query Tool，输入对应的 SQL 语句，单击运行，刷新数据库，查看 aehero_name 字段的属性，字段类型已变为 text。

源代码：

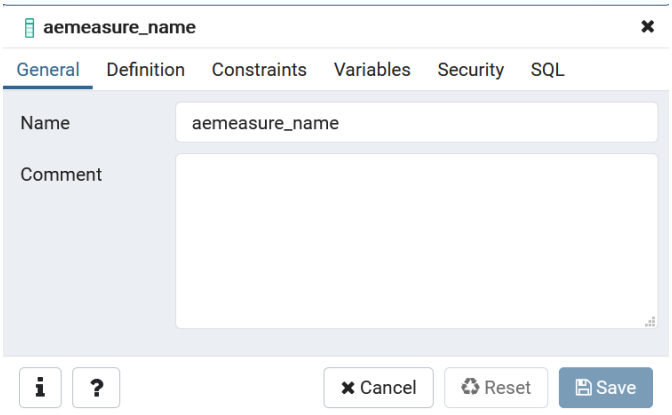
--修改数据表 hero1 中的 aehero_name 字段的类型为文本类型

alter table hero1 alter column aehero_name type text



(6) 使用 SQL 语句把数据表 aemeasure 中的 aemeasure_name 改为 aemeasurename。

修改前 aemeasure_name 字段名称如下图所示。

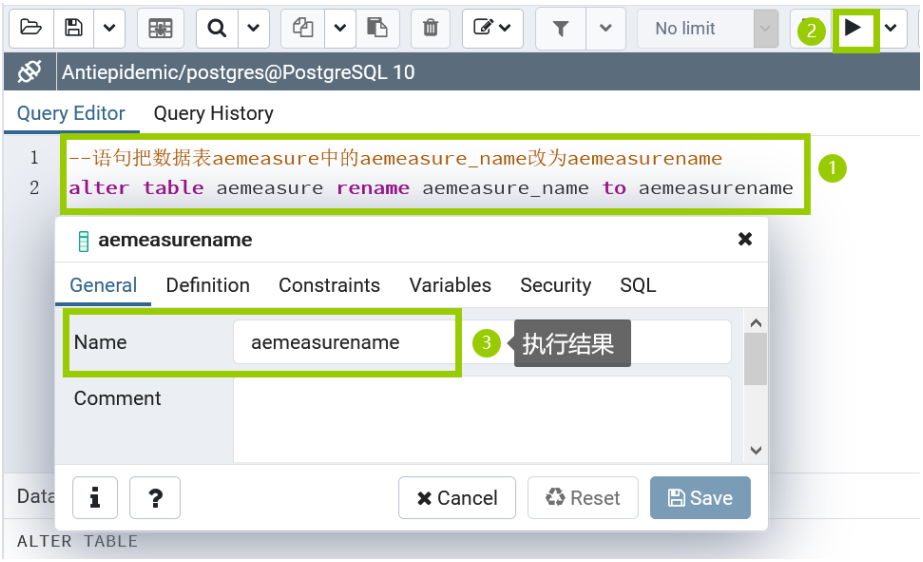


打开 Antiepidemic 数据库的 Query Tool，输入 SQL 语句，单击执行，刷新后查看该字段名称变化，修改成功。源代码截图、步骤截图和执行结果如下图所示。

源代码：

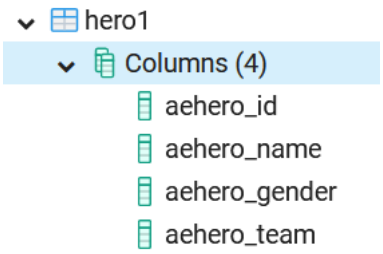
--语句把数据表 aemeasure 中的 aemeasure_name 改为 aemeasurename

alter table aemeasure rename aemeasure_name to aemeasurename



(7) 使用 SQL 语句删除数据表 hero1 中的字段 aehero_gender。

操作执行前数据表 hero1 中有四个字段，其中包含 aehero_gender

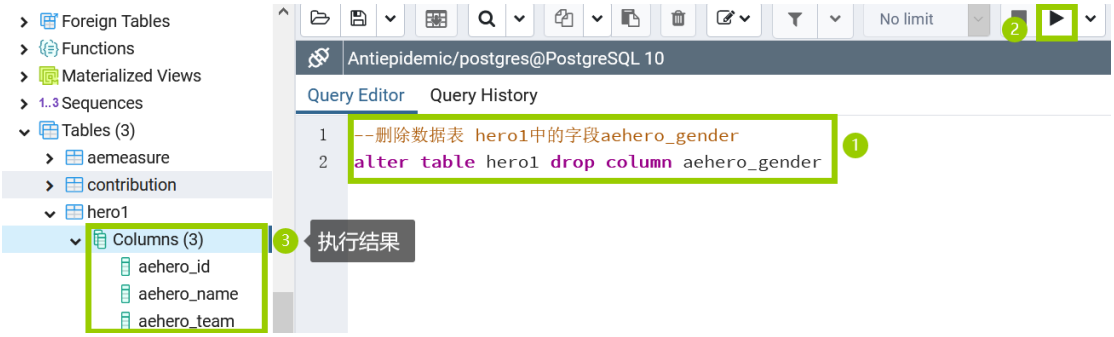


打开 Antiepidemic 数据库的 Query Tool，输入对应的 SQL 语句，单击运行，刷新查看数据表 hero1 的字段，成功删除 aehero_gender。

源代码：

--删除数据表 hero1 中的字段 aehero_gender

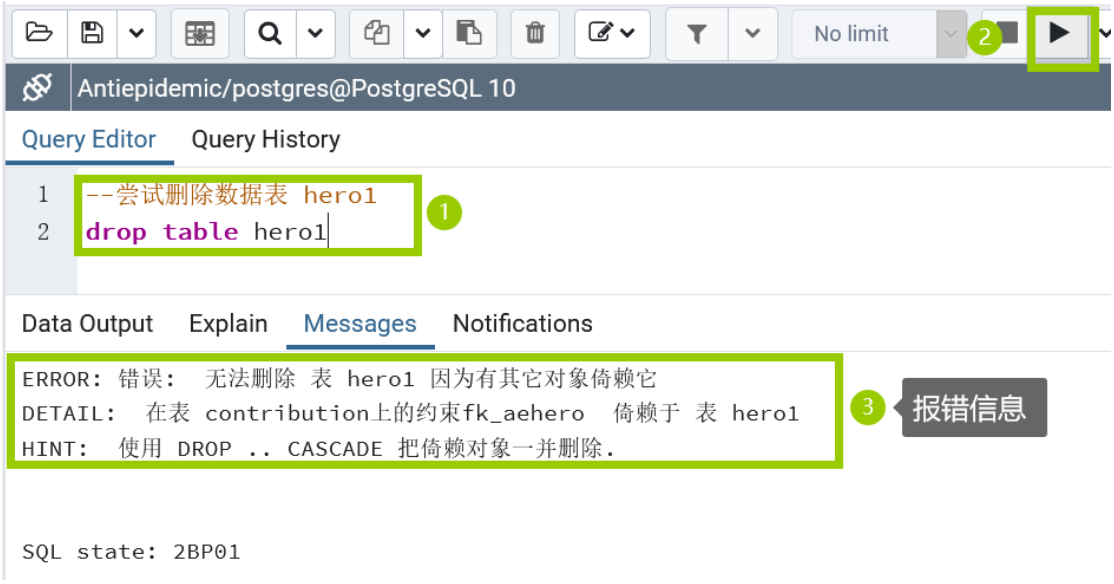
alter table hero1 drop column aehero_gender



(8) 使用 SQL 语句删除数据表 hero1，这时会发生什么情况，截图说明。

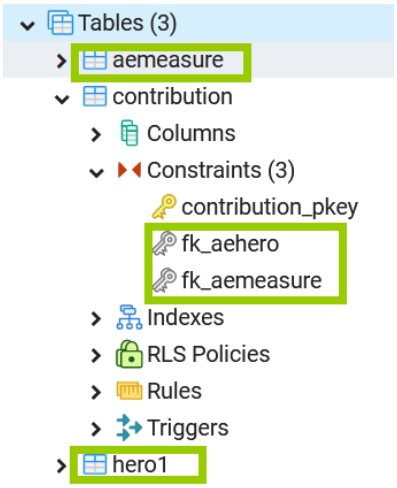
在 Query Tool 工具中输入相应 SQL 语句尝试删除数据表 hero1，单击运行，出现报错。

原因是存在依赖于 hero1 中字段的外键，数据表 contribution 的外键 fk_aehero 依赖于 hero1.aehero_id。



(9) 使用 SQL 语句删除数据表 hero1，aemeasure。

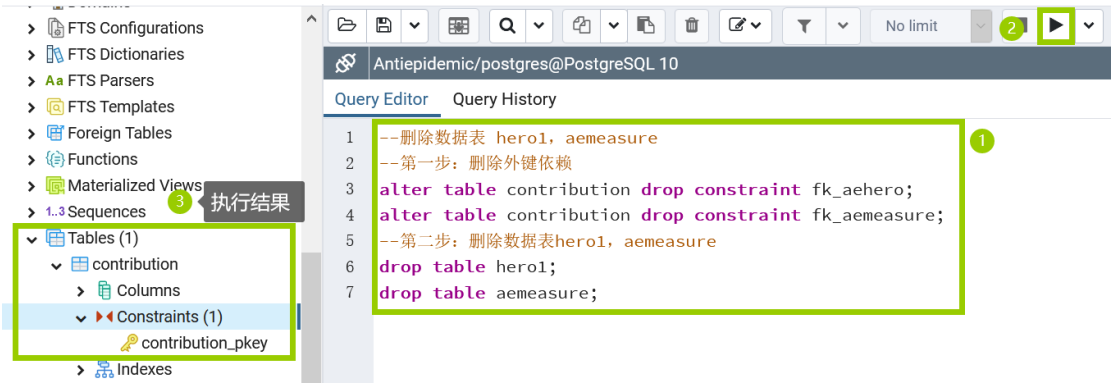
因为这两个数据表都有外键依赖于它，因此需要先删除依赖于它们的外键。操作前状态如图所示。



在 Antiepidemic 数据库的 Query Tool 中，输入对应的 SQL 语句，单击运行，刷新后发现数据表 hero1、aemeasure 已经删除，依赖于这两个表的外键同样已经删除。

源代码：

```
--删除数据表 hero1，aemeasure
--第一步：删除外键依赖
alter table contribution drop constraint fk_aehero;
alter table contribution drop constraint fk_aemeasure;
--第二步：删除数据表 hero1，aemeasure
drop table hero1;
drop table aemeasure;
```



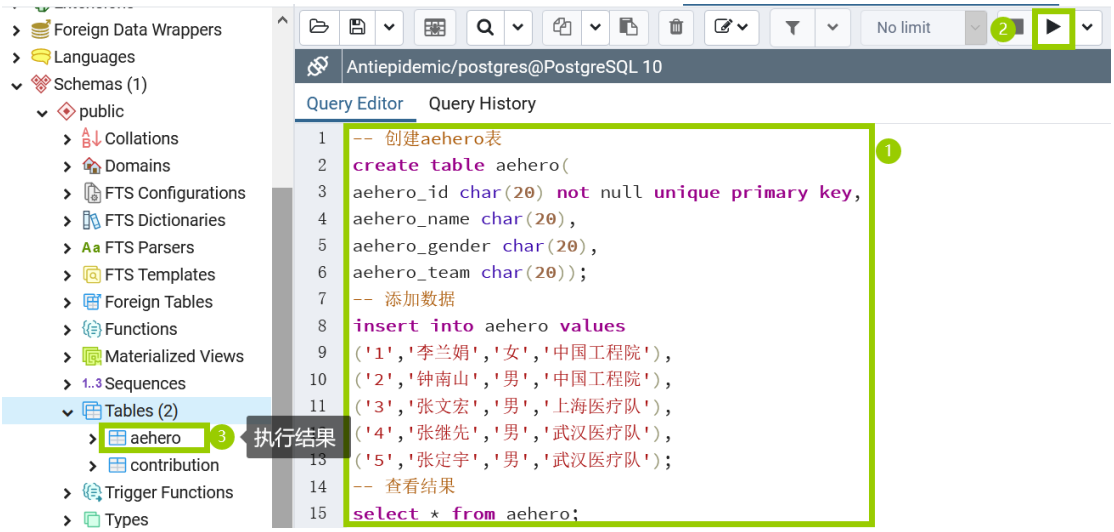
(10)创建如下 aehero 表，使用 SQL 语句向 aehero 表中插入数据:

aehero_id	aehero_name	aehero_gender	aehero_team
1	李兰娟	女	中国工程院
2	钟南山	男	中国工程院
3	张文宏	男	上海医疗队
4	张继先	男	武汉医疗队
5	张定宇	男	武汉医疗队

首先使用 SQL 语句创建 aehero 数据表，再使用 insert into 语句同时向表中插入多条数据，最后展示表中所有数据。

源代码：

```
-- 创建 aehero 表
create table aehero(
aehero_id char(20) not null unique primary key,
aehero_name char(20),
aehero_gender char(20),
aehero_team char(20));
-- 添加数据
insert into aehero values
('1','李兰娟','女','中国工程院'),
('2','钟南山','男','中国工程院'),
('3','张文宏','男','上海医疗队'),
('4','张继先','男','武汉医疗队'),
('5','张定宇','男','武汉医疗队');
-- 查看结果
select * from aehero;
```



Data Output | Explain | Messages | Notifications

	aehero_id [PK] character (20)	aehero_name character (20)	aehero_gender character (20)	aehero_team character (20)
1	1	李兰娟	女	中国工程院
2	2	钟南山	男	中国工程院
3	3	张文宏	男	上海医疗队
4	4	张继先	男	武汉医疗队
5	5	张定宇	男	武汉医疗队

(11)创建如下 aemeasure 表，使用 SQL 语句向 aemeasure 表中插入数据:

aemeasure_id	aemeasure_name	aemeasure_detail
1	封锁城市	武汉市关闭离汉通道，暂停市内交通，取消社会聚集活动，严格佩戴口罩。
2	分离病毒	当人类拥有了分离后的毒株，才能对人类进行临床实验，尽可能筛选出合适的治疗药物和治疗手段。
3	研制中医药剂	面对疫情，抗疫英雄们不分日夜，潜心研究，治愈患者成果显著。

首先使用 SQL 语句创建 aemeasure 数据表，再使用 insert into 语句同时向表中插入多条数据，最后展示表中所有数据。

源代码：

```
--首先创建如下 aemeasure 表:
create table aemeasure(
aemeasure_id char(20) not null unique primary key,
aemeasure_name char(20),
aemeasure_detail char(100));
--使用 SQL 语句向 aemeasure 表中插入数据
insert into aemeasure values
('1','封锁城市','武汉市关闭离汉通道，暂停市内交通，取消社会聚集活动，严格佩戴
```

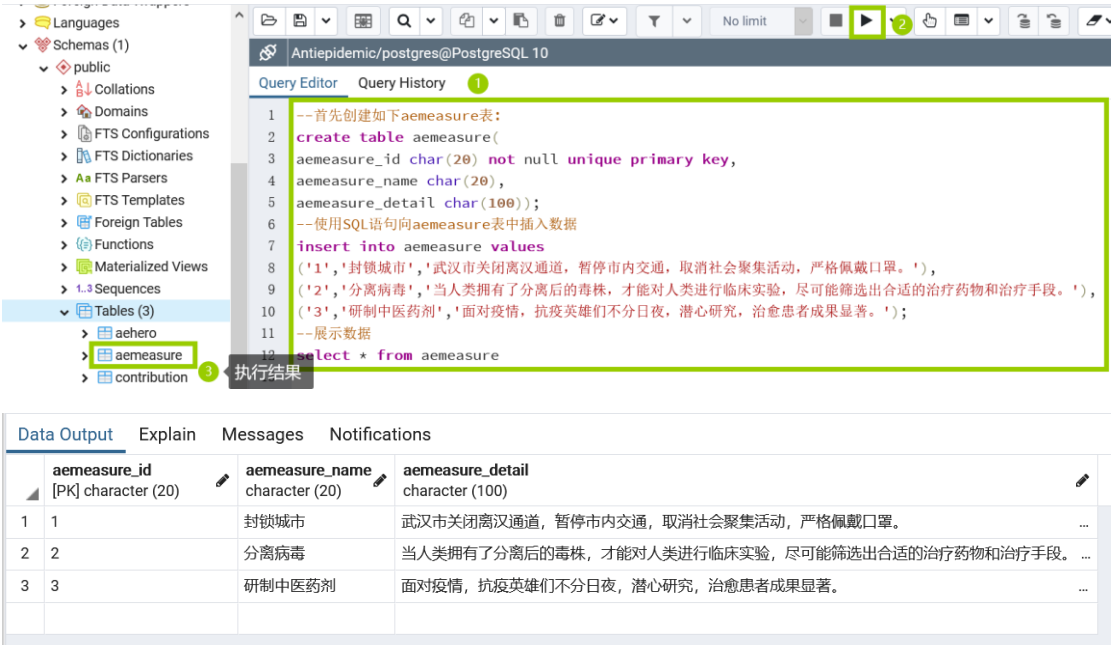
口罩。'),

('2','分离病毒','当人类拥有了分离后的毒株，才能对人类进行临床实验，尽可能筛选出合适的治疗药物和治疗手段。'),

('3','研制中医药剂','面对疫情，抗疫英雄们不分日夜，潜心研究，治愈患者成果显著。');

--展示数据

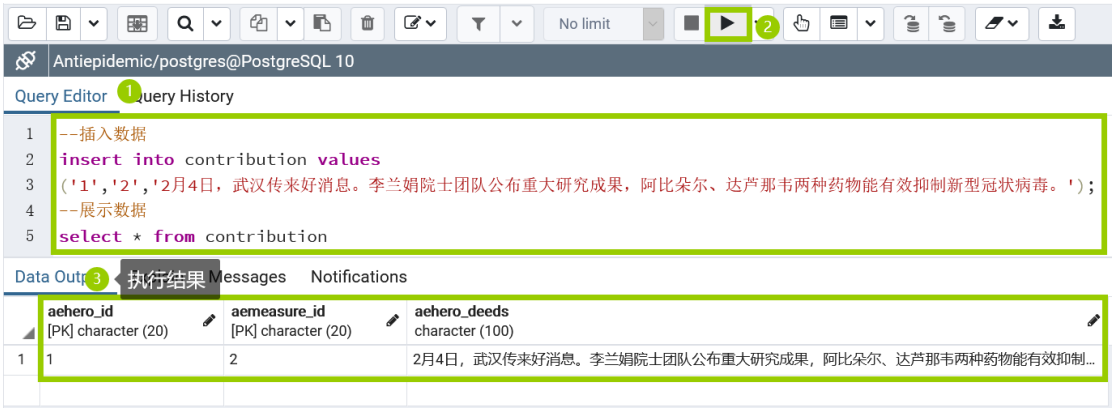
select * from aemeasure;



(12)使用 SQL 语句向 contribution 表中插入如下指定字段的数据：

aehero_id	aemeasure_id	aehero_deeds
1	3	2 月 4 日，武汉传来好消息。李兰娟院士团队公布重大研究成果，阿比朵尔、达芦那韦两种药物能有效抑制新型冠状病毒。

使用 insert into 语句向 contribution 表中插入数据，再展示数据



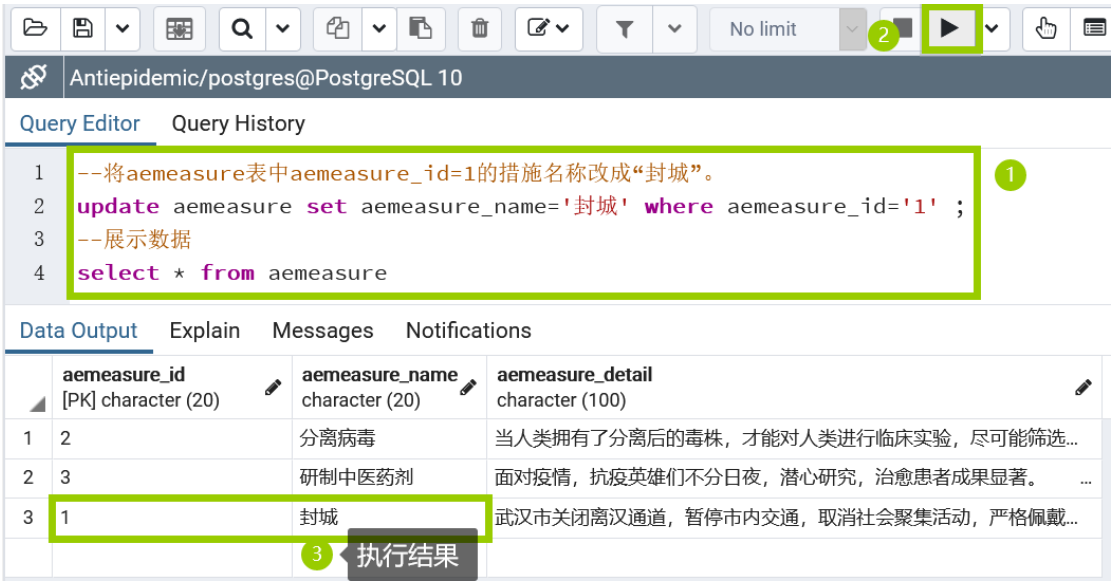
(13)使用 SQL 语句将 aemeasure 表中 aemeasure_id=1 的措施名称改成“封城”。

修改前状态如下：

使用 SQL 语句对 aemeasure 表进行修改，单击运行，展示修改后数据。

源代码：

```
--将 aemeasure 表中 aemeasure_id=1 的措施名称改成“封城”。
update aemeasure set aemeasure_name='封城' where aemeasure_id='1' ;
--展示数据
select * from aemeasure
```



(14)使用 SQL 语句将 contribution 表中所有数据删除。

操作前数据表 contribution 中数据见（12）截图。

使用 SQL 语句删除表中所有数据，单击运行，可见现在表中数据为空。

源代码：

```
--使用 SQL 语句将 contribution 表中所有数据删除
delete from contribution;
--展示数据
select * from contribution
```


No limit
2
Antiepidemic/postgres@PostgreSQL 10
Query Editor
Query History
1 --使用SQL语句将contribution表中所有数据删除
2 delete from contribution;
3 --展示数据
4 select * from contribution
1
Data Output
Explain
Messages
Notifications
aehero_id
[PK] character (20)
aemeasure_id
[PK] character (20)
aehero_deeds
character (100)
3
执行结果