**福医服务器 构建PostgreSQL服务 以及 数据使用说明**

**一、说明：**

**1、此服务依托于何俊老师的服务器账号（hej@210.34.96.245），后续需要增加数据库需向何老师问询账号；**

**2、这个文档包括，下载安装postgresql 软件，挂载服务，导入MIMIC-IV（verson 2.0） 原始数据并构建其衍生表格。**

**3、其他数据库可参考此文档及其官方文档。**

**4、文档后部分有使用说明，及使用规范说明。**

**PS： 使用 Navicat，请使用**

**账号：postgres**

**密码：pgs.123456.**

**因为都是登录同一账号，所以 可以操作 其他 user 的模式，请谨慎操作！**

**二、PostgerSQL安装及配置（之后直接添加数据库，此步骤不需要重新运行）**

1. **conda 安装PostgerSQL（base环境安装，未创建新环境）：**

# 寻找版本

conda search postgresql

# 安装

conda install postgresql=12.9

# 创建目录 (存放数据库数据的目录)

cd ~

mkdir PostgreSQL

cd PostgreSQL/

mkdir pgsql

mkdir pgsql/data

cd ~/ PostgreSQL/ pgsql/data

# 创建一个新的数据库集簇。

initdb ~/PostgreSQL/pgsql/data

# 将在后台启动服务器并且把输出放到指定的日志文件中

## 使用qsub 将服务挂在在 非管理节点 下，这里选择 cu01 计算节点

qsub -I -l nodes=cu01

## 启动服务

pg\_ctl -D ~/PostgreSQL/pgsql/data -l logfile start

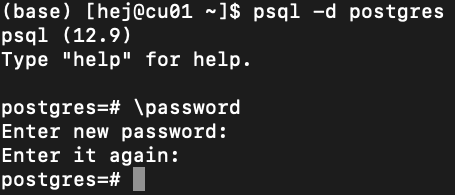
**2、配置 postgresql**

# 进入默认数据库 (需qsub 到cu01节点运行)

psql -d postgres

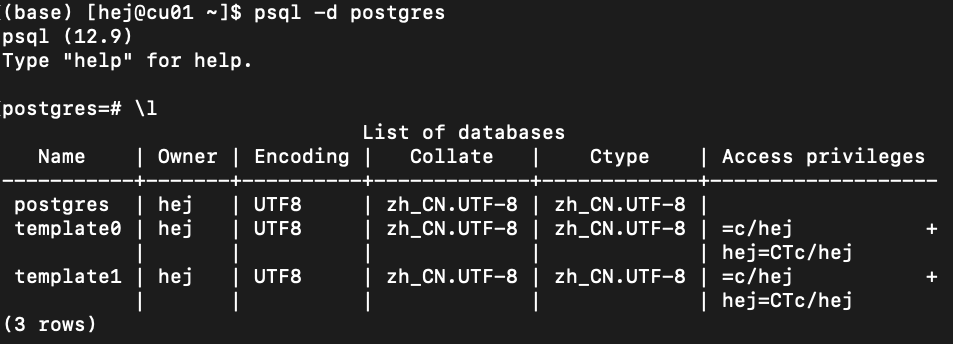
# 修改密码 ( 请勿修改，密码请问 何俊老师 )

# \password



# \l 查看当前数据库

\l

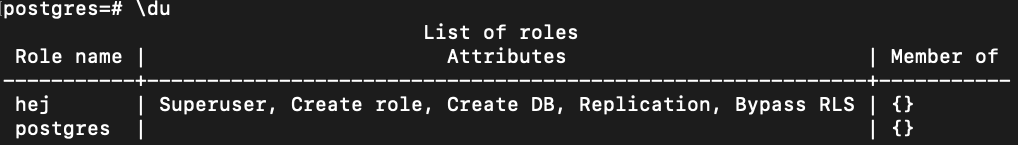


# 创建新用户（用户名postgres , 密码: pgs.123456.）

create user postgres with password 'pgs.123456.';

# 查看 有哪些用户

\du



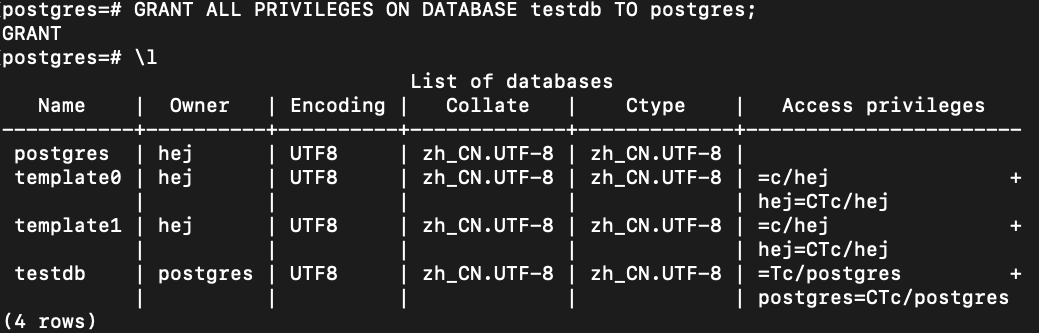
# 创建新数据库 ( testdb )

CREATE DATABASE testdb OWNER postgres;



# 将新数据库关联用户

GRANT ALL PRIVILEGES ON DATABASE testdb TO postgres;



# 退出 postgresql

\q



# 配置远程登录

## 配置postgresql.conf 文件，

## 添加/修改：在所有IP地址上监听，从而允许远程连接到数据库服务器：

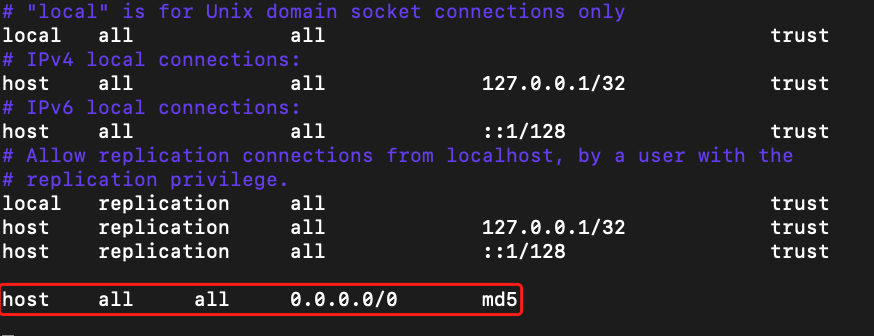
vim ~/PostgreSQL/pgsql/data/postgresql.conf



## 配置pg\_hba.conf 文件，

## 添加/修改：允许任意用户从任意机器上以密码方式访问数据库，把下行添加为第一条规则：

vim ~/PostgreSQL/pgsql/data/pg\_hba.conf

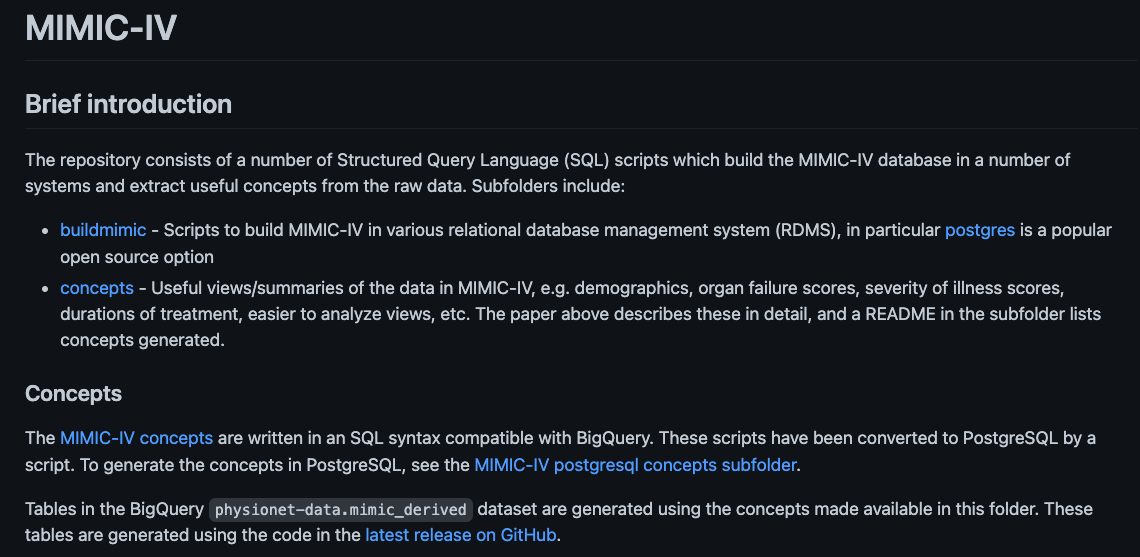


host    all             all             0.0.0.0/0               md5

**三、导入HER数据至postgresql**

**1、MIMIC-IV数据导入postgresql 服务器，并根据官方代码获取衍生表格（derived tables）**

**这个部分以构建MIMIC-IV version 2.0 为例！**



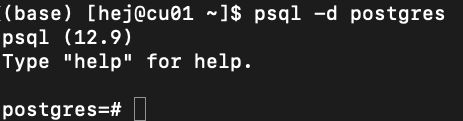
(<https://github.com/MIT-LCP/mimic-code/tree/main/mimic-iv/>)

# 进入何老师账号，并通过 qsub进入运行服务的节点。

qsub -I -l nodes=cu01

# 进入postgresql数据库 (默认用户: hej (拥有最高权限))

psql -d postgres



# 接下来导入数据（也可根据官方文档配置）

官网代码地址（包括创建表格结构、导入数据等sql代码，）：

https://github.com/MIT-LCP/mimic-code/tree/main/mimic-iv

# 创建数据库：mimiciv\_20

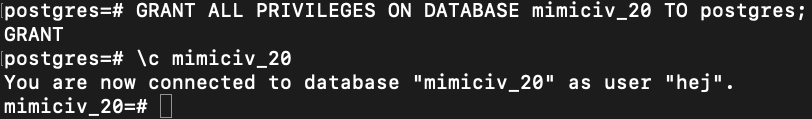
# 创建其他数据库需要 **更改名字**，不换名字mimiciv\_20会被清空！

DROP DATABASE IF EXISTS mimiciv\_20;

CREATE DATABASE mimiciv\_20 OWNER hej;

# 进入数据库

\c mimiciv\_20



# 创建表格 (绝对路径)

\i /public/hej/PostgreSQL/EHR\_RawData/Code\_sql/mimic-code-main/mimic-iv/buildmimic/postgres/create.sql

# 设置数据路径

\set mimic\_data\_dir '/public/hej/PostgreSQL/EHR\_RawData/MIMIC\_IV/mimiciv\_2.0'

**##** mimic\_data\_dir 这个变量是load\_gz.sql里面需要用到的变量，其他的数据库根据情况处理。

\encoding 'UTF8'

# 导入数据 (gz文件，不解压)

\i /public/hej/PostgreSQL/EHR\_RawData/Code\_sql/mimic-code-main/mimic-iv/buildmimic/postgres/load\_gz.sql

# Creating Primary Keys

\i /public/hej/PostgreSQL/EHR\_RawData/Code\_sql/mimic-code-main/mimic-iv/buildmimic/postgres/constraint.sql

# Indexes for all MIMIC-IV modules

\i /public/hej/PostgreSQL/EHR\_RawData/Code\_sql/mimic-code-main/mimic-iv/buildmimic/postgres/index.sql

# 衍生表格构建（derived table）

# 设置路径

show search\_path;

set search\_path to mimiciv\_derived;

# 载入自定函数（只需运行一次）

\i /public/hej/PostgreSQL/EHR\_RawData/Code\_sql/mimic-code-main/mimic-iv/concepts/postgres/postgres-functions.sql

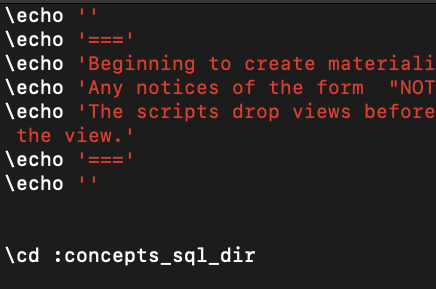
# 设置路径

\set concepts\_sql\_dir '/public/hej/PostgreSQL/EHR\_RawData/Code\_sql/mimic-code-main/mimic-iv/concepts/postgres'

## 需要对postgres-make-concepts.sql修改

## 加入一行 如下图

# \cd :concepts\_sql\_dir



# make concepts 生成衍生表 （有些代码有问题，我已经修改，放在/public/hej/PostgreSQL/EHR\_RawData/Code\_sql/mimic-code-main/mimic-iv/concepts/postgres\_my文件夹下）

\i /public/hej/PostgreSQL/EHR\_RawData/Code\_sql/mimic-code-main/mimic-iv/concepts/postgres/postgres-make-concepts.sql

# 设置 用户postgres只读权限

\c mimiciv\_20

GRANT ALL PRIVILEGES ON DATABASE mimiciv\_20 TO postgres;

# 模式 mimiciv\_derived

GRANT usage ON SCHEMA mimiciv\_derived TO postgres;

GRANT SELECT ON ALL tables IN SCHEMA mimiciv\_derived TO postgres;

# 模式 mimiciv\_hosp

GRANT usage ON SCHEMA mimiciv\_hosp TO postgres;

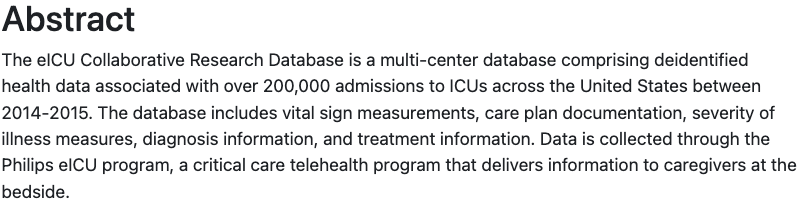
GRANT SELECT ON ALL tables IN SCHEMA mimiciv\_hosp TO postgres;

# 模式 mimiciv\_icu

GRANT usage ON SCHEMA mimiciv\_icu TO postgres;

GRANT SELECT ON ALL tables IN SCHEMA mimiciv\_icu TO postgres;

**2、eICU(version 2.0)数据导入postgresql 服务器，并根据官方代码获取衍生表格（derived tables）**



（<https://physionet.org/content/eicu-crd/2.0/>）

（<https://github.com/mit-lcp/eicu-code>）

# 进入何老师账号，并通过 qsub进入运行服务的节点。

qsub -I -l nodes=cu01

# 进入postgresql数据库 (默认用户: hej (拥有最高权限))

psql -d postgres

# 接下来导入数据（也可根据官方文档配置）

官网代码地址（包括创建表格结构、导入数据等sql代码，）：

https://github.com/MIT-LCP/mimic-code/tree/main/mimic-iv

# 创建数据库：eicu\_20

# 创建其他数据库需要 **更改名字**，不换名字eicu\_20会被清空！

DROP DATABASE IF EXISTS eicu\_20;

CREATE DATABASE eicu\_20 OWNER hej;

# 进入数据库

\c eicu\_20

# 设置模式及进入该模式

show search\_path;

set search\_path to eicu\_crd; # 没有eicu\_crd则不会有 set 打印，需要创建一个

# create schema eicu\_crd;



# 创建表格 (绝对路径)

\i /public/hej/PostgreSQL/EHR\_RawData/Code\_sql/eicu-code-master/build-db/postgres/postgres\_create\_tables.sql

# 设置数据路径

\set datadir '/public/hej/PostgreSQL/EHR\_RawData/eicu-collaborative-research-database-2.0'

## datadir这个变量是load\_gz.sql里面需要用到的变量，其他的数据库根据情况处理。

\encoding 'UTF8'

# 导入数据 (gz文件，不解压)

\i /public/hej/PostgreSQL/EHR\_RawData/Code\_sql/eicu-code-master/build-db/postgres/postgres\_load\_data\_gz.sql

# Creating Primary Keys

\i /public/hej/PostgreSQL/EHR\_RawData/Code\_sql/eicu-code-master/build-db/postgres/postgres\_add\_constraints.sql

# Indexes for all MIMIC-IV modules

\i /public/hej/PostgreSQL/EHR\_RawData/Code\_sql/eicu-code-master/build-db/postgres/postgres\_add\_indexes.sql

# 衍生表格构建（derived tables）

create schema eicu\_derived;

# 设置路径

show search\_path;

set search\_path to eicu\_crd;

# 载入自定函数（只需运行一次）（使用mimic的）

\i /public/hej/PostgreSQL/EHR\_RawData/Code\_sql/mimic-code-main/mimic-iv/concepts/postgres/postgres-functions.sql

# 设置路径 （有的代码修改了，存在concepts\_my文件夹）

\set concepts\_sql\_dir '/public/hej/PostgreSQL/EHR\_RawData/Code\_sql/eicu-code-master/concepts\_my'

## 需要自己创建 postgres-make-concepts.sql 文件

## 并加入以下一行命令

## \cd :concepts\_sql\_dir

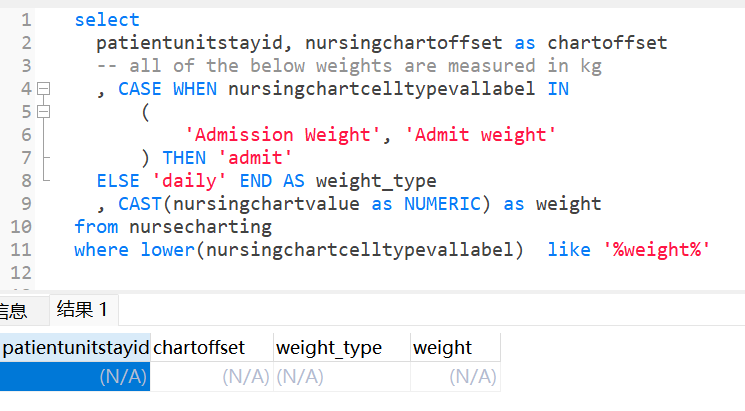
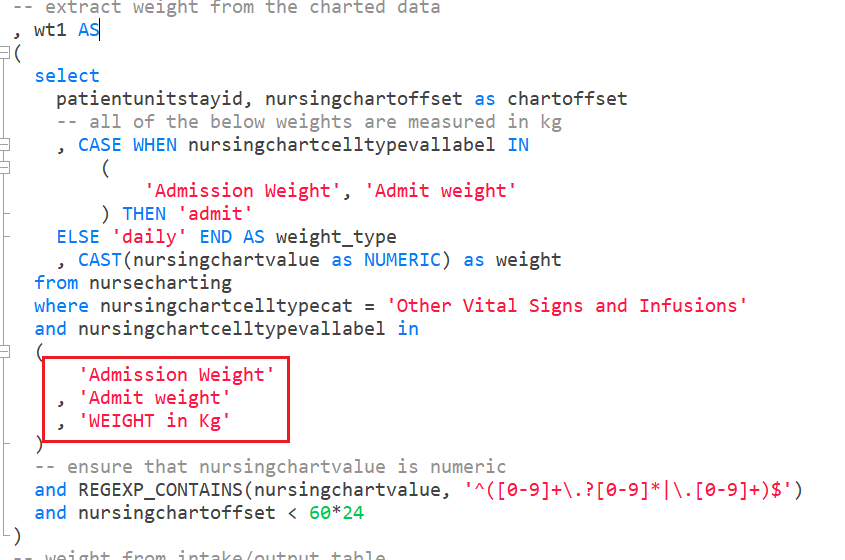
# make concepts 生成衍生表 （有些代码有问题，我已经修改，放在/public/hej/PostgreSQL/EHR\_RawData/Code\_sql/eicu-code-master/concepts\_my文件夹下）

\i /public/hej/PostgreSQL/EHR\_RawData/Code\_sql/eicu-code-master/concepts\_my /postgres-make-concepts.sql

ps: pivoted-weight.sql 报错，后查询得知，nursecharting 中没有记录weight信息。

但是 eicu\_crd.patient 以及 eicu\_derived.icustay\_detail的里面有。





# 设置 用户postgres只读权限

\c eicu\_20

GRANT ALL PRIVILEGES ON DATABASE eicu\_20 TO postgres;

# 模式 eicu\_crd

GRANT usage ON SCHEMA eicu\_crd TO postgres;

GRANT SELECT ON ALL tables IN SCHEMA eicu\_crd TO postgres;

# 模式 eicu\_derived

GRANT usage ON SCHEMA eicu\_derived TO postgres;

GRANT SELECT ON ALL tables IN SCHEMA eicu\_derived TO postgres;

**四、本地连接 postgresql 服务器**

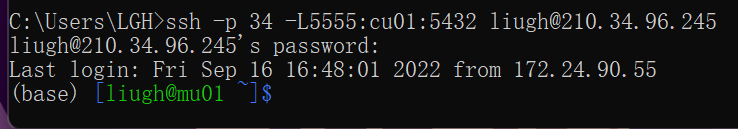
1、由于服务是运行在cu01 节点上，本地直接ping不通，所以需要做个隧道进行端口映射，两种连接方式：

* 1. window电脑使用cmd或者powershell，ssh连接 （mac 直接 在终端里运行ssh命令）

# 如我的账号是liugh, 可以直接ssh连接

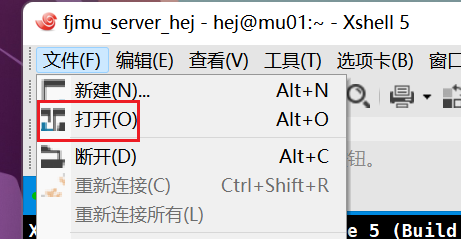
## -L 参数 将服务器 cu01:5432 端口映射到 本地5555端口，构成隧道。

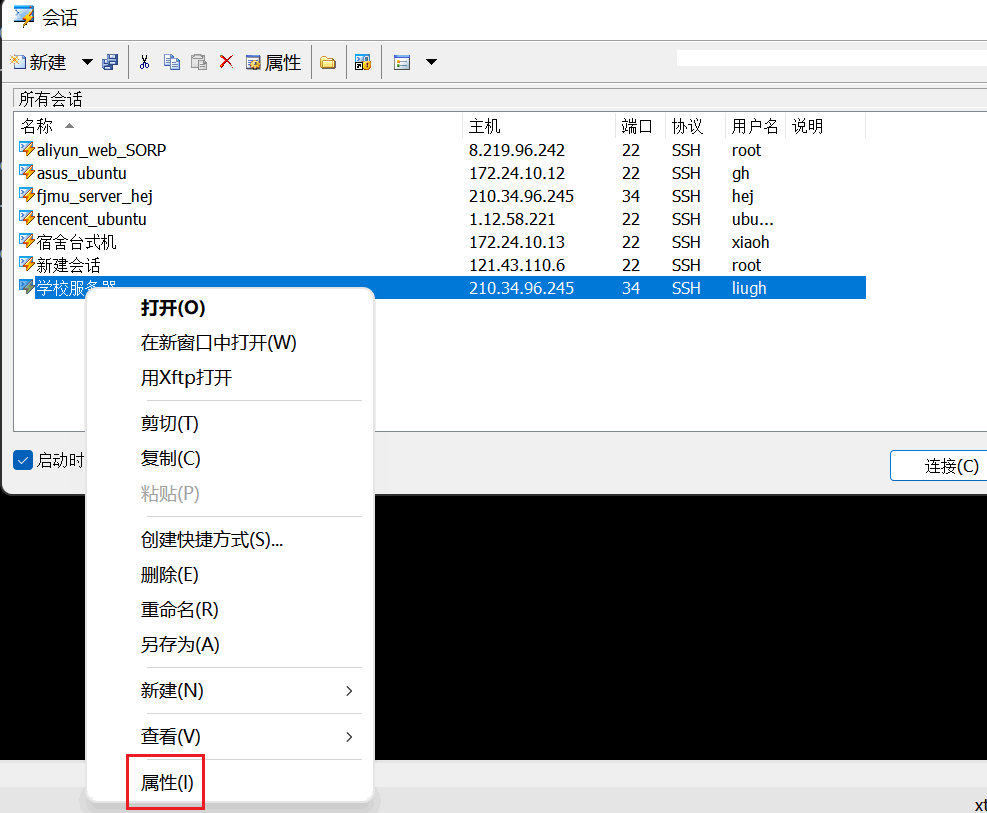
ssh -p 34 -L5555:cu01:5432 liugh@210.34.96.245



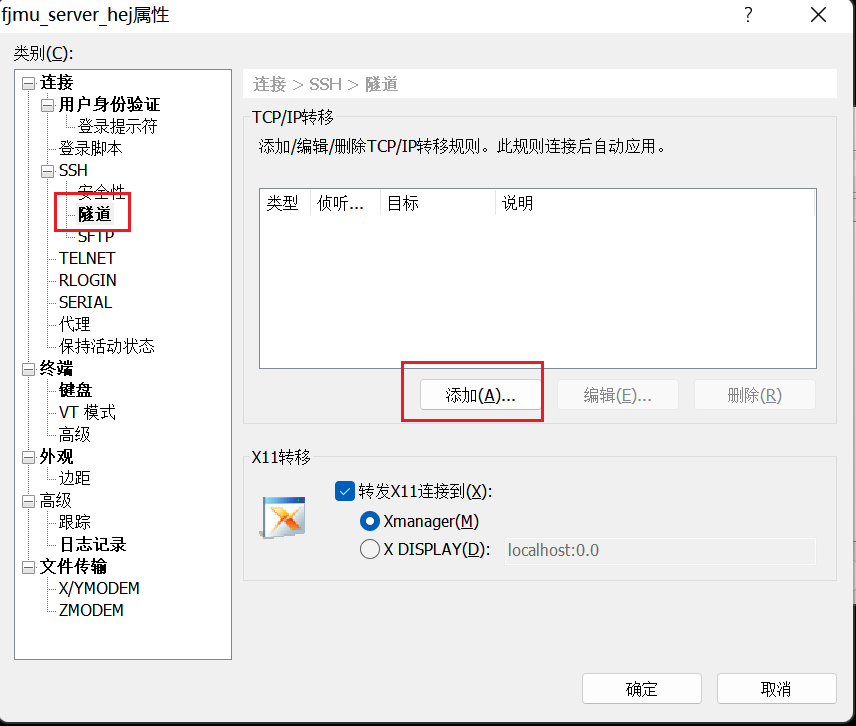
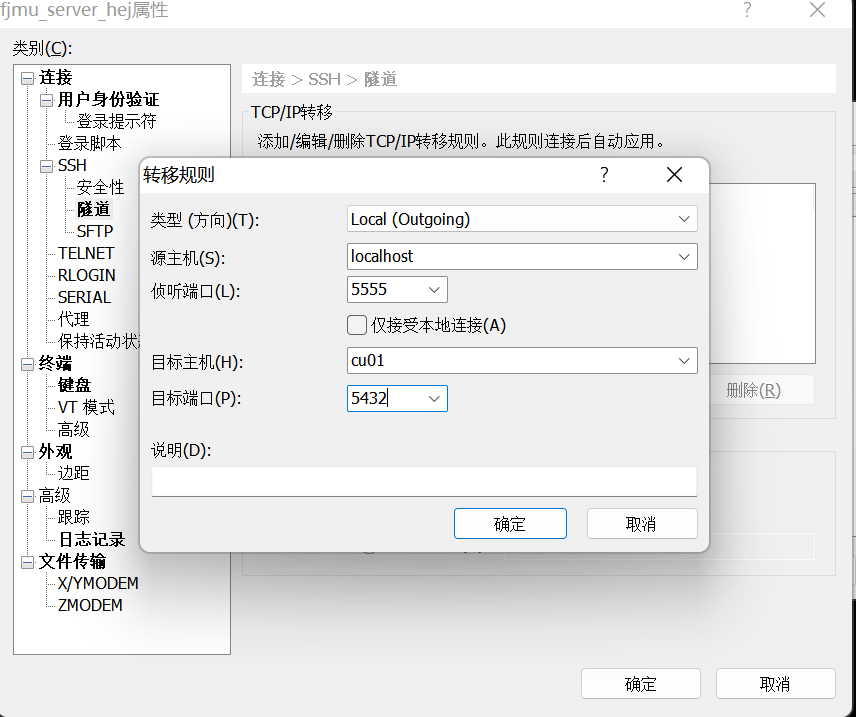
* 1. 使用xshell, 构建隧道

# 打开服务器，右键选择属性配置





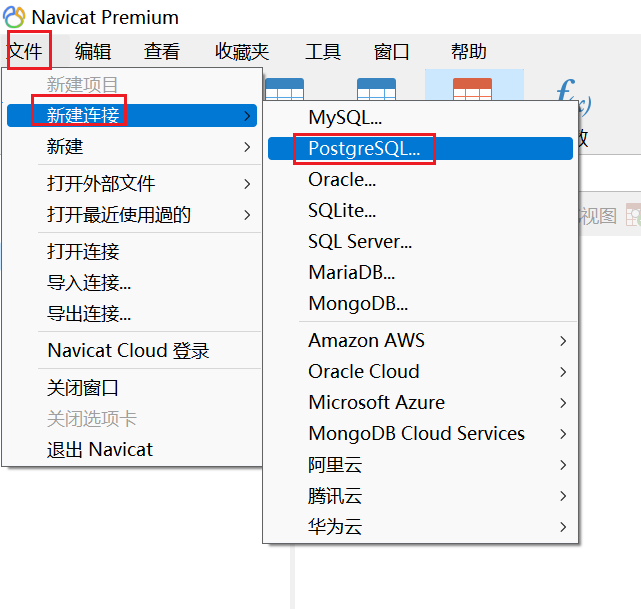
# 选择 隧道，然后添加端口，如下图配置

配置之后，登录服务器会自动映射端口形成隧道，映射到本地。

1. 使用Navicat 连接数据库。

# 新建连接，如下图：

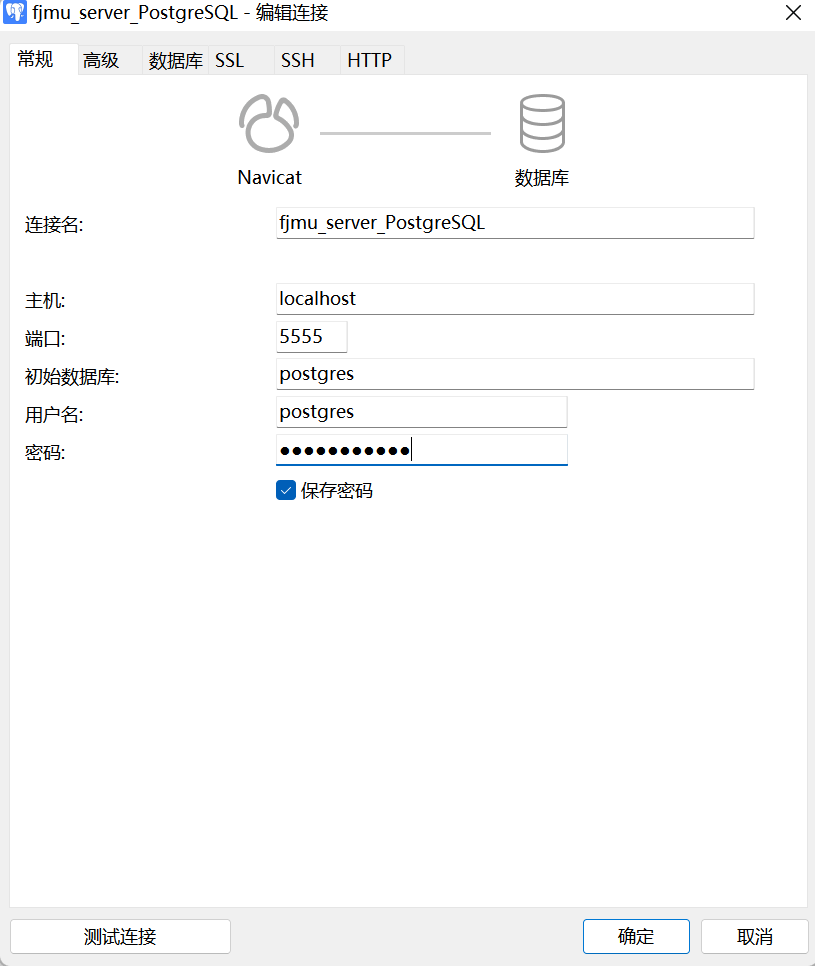


# 配置，如下图：

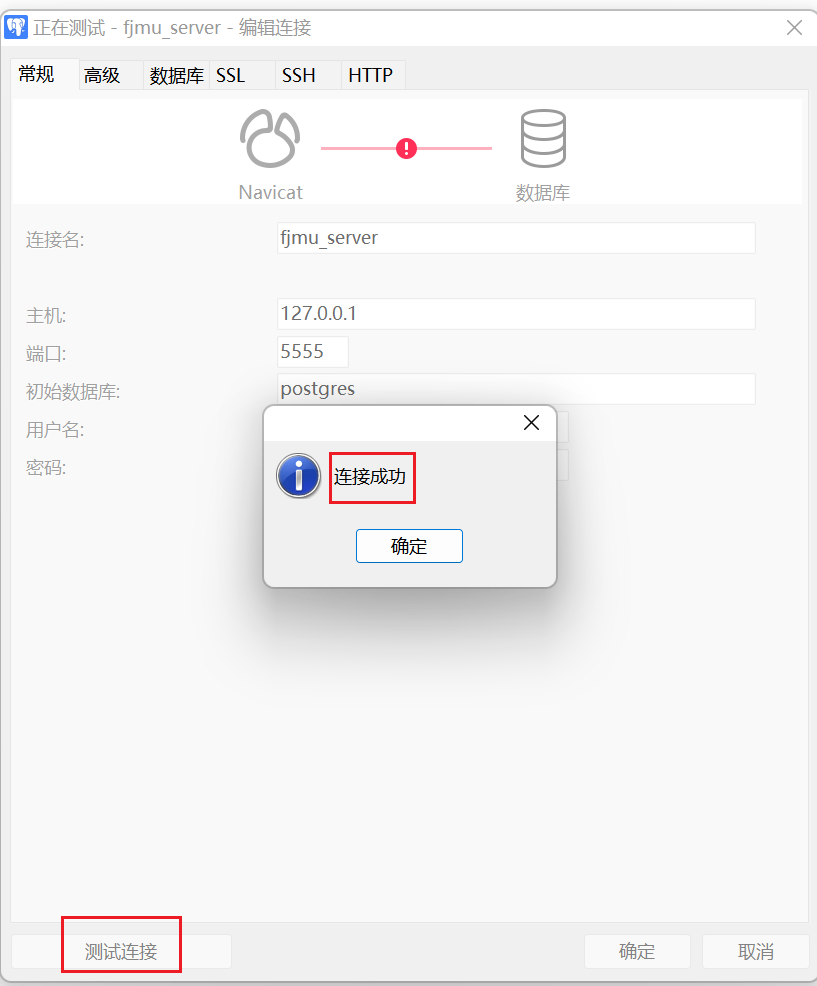
## 因为上面做了隧道进行了端口映射，所以直接连接本地的5555端口就是连接服务器cu01:5432端口。

## 账号: postgres

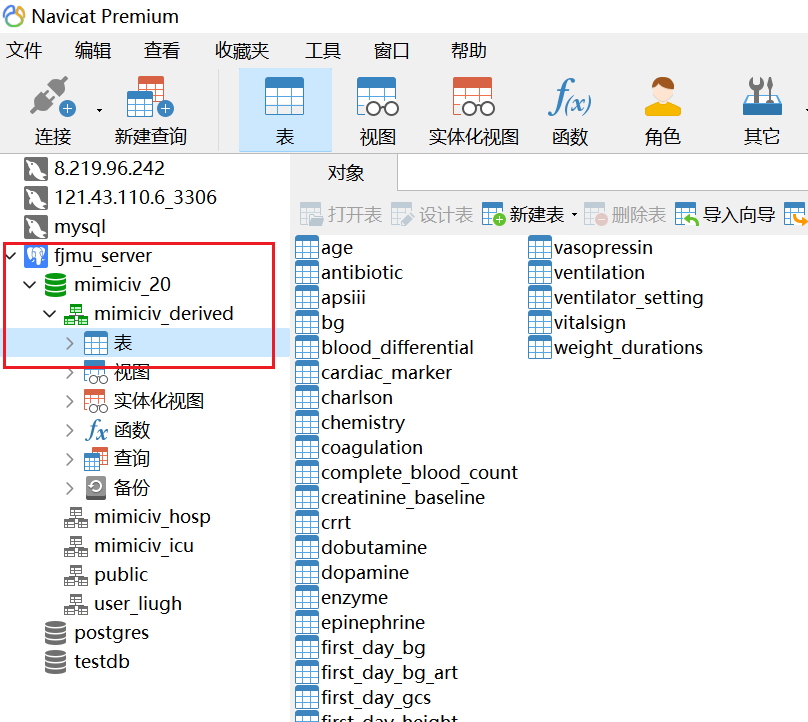
## 密码: pgs.123456.



# 测试连接



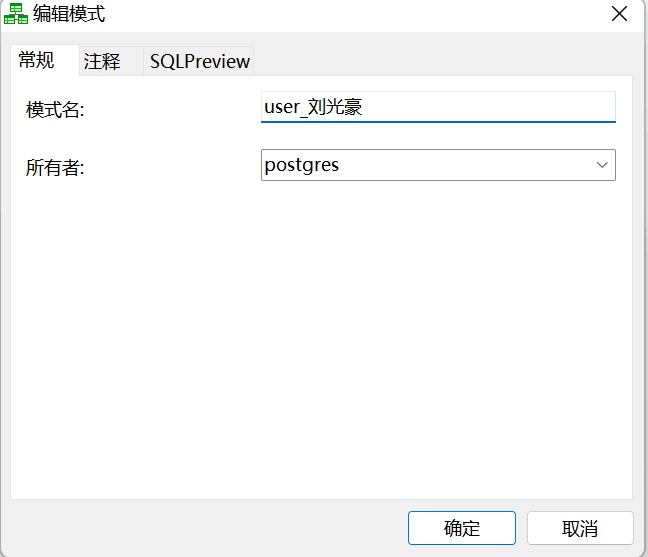
# 如下图，成功连接并显示数据库：



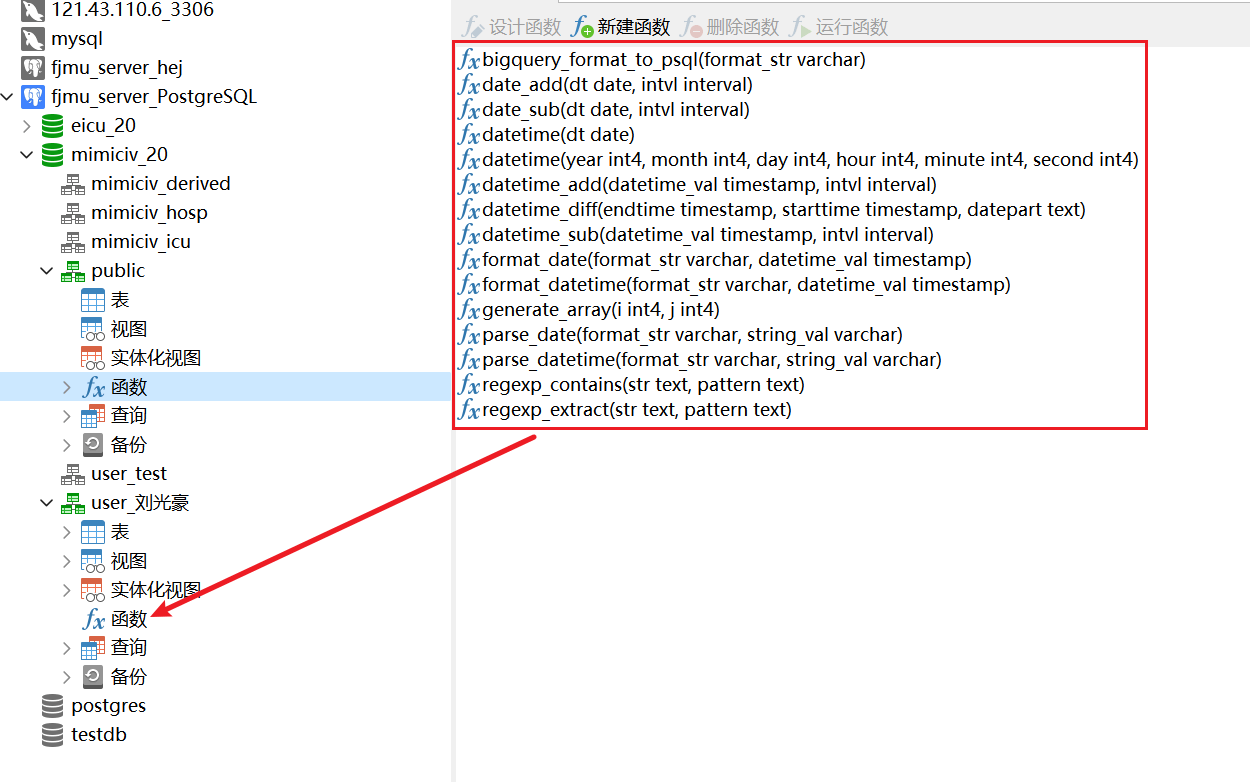
**五、个人使用**

需要根据情况个性化提取数据，需要建立自己的 个人 模式 如：

**模式名统一设置： user\_名字，如下图**

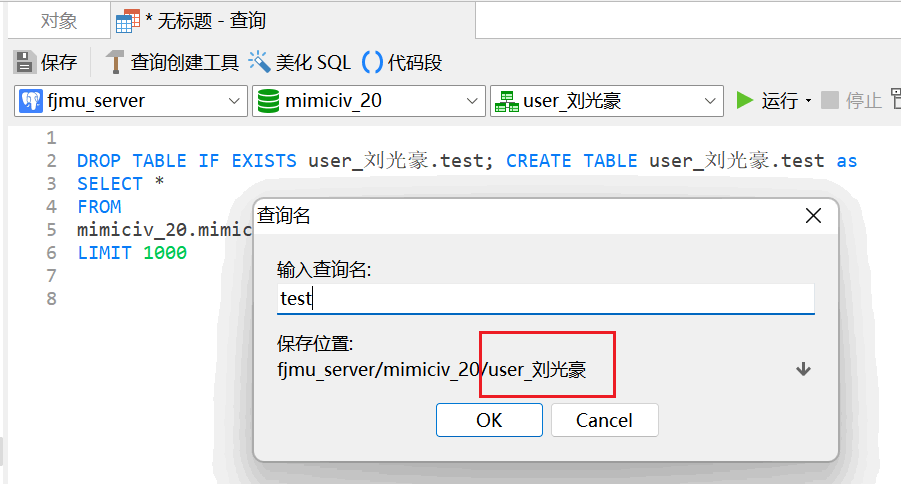
**需要的函数可以从 模式 public 复制到自己模式的函数下：**

****

**请务必将新创建的表，都存于自己的模式中，如：**

****

**保存 查询代码 也存于自己的模式中：**

****

# 衍生表中的 查询代码中有一个readme

# 里面记录着各个表的类别或归属于哪一类

