

## 1 安装包准备

1) 将资料里 mysql 文件夹及里面所有内容上传到/opt/software/mysql 目录下

```
[iflytek@hadoop102~]# mkdir /opt/software/mysql
[iflytek@hadoop102 software]$ cd /opt/software/mysql/

install_mysql.sh
mysql-community-client-8.0.31-1.el7.x86_64.rpm
mysql-community-client-plugins-8.0.31-1.el7.x86_64.rpm
mysql-community-common-8.0.31-1.el7.x86_64.rpm
mysql-community-icu-data-files-8.0.31-1.el7.x86_64.rpm
mysql-community-libs-8.0.31-1.el7.x86_64.rpm
mysql-community-libs-compat-8.0.31-1.el7.x86_64.rpm
mysql-community-server-8.0.31-1.el7.x86_64.rpm
mysql-connector-j-8.0.31.jar
```

## 2 安装 MySQL

1) 如果是阿里云服务器按照如下步骤执行

说明：由于阿里云服务器安装的是 Linux 最小系统版，没有如下工具，所以需要安装。

(1) 卸载 MySQL 依赖，虽然机器上没有装 MySQL，但是这一步不可少

```
[iflytek@hadoop102 mysql]# sudo yum remove mysql-libs
```

(2) 下载依赖并安装

```
[iflytek@hadoop102 mysql]# sudo yum install libaio
[iflytek@hadoop102 mysql]# sudo yum -y install autoconf
```

2) 切换到 hadoop102 的 root 用户

```
[iflytek@hadoop102 mysql]$ su root
```

3) 执行/opt/software/mysql/目录下 install\_mysql.sh

```
[root@hadoop102 mysql]# vim install_mysql.sh

#!/bin/bash
set -x
[ "$(whoami)" = "root" ] || exit 1
[ "$(ls *.rpm | wc -l)" = "7" ] || exit 1
test -f mysql-community-client-8.0.31-1.el7.x86_64.rpm && \
test -f mysql-community-client-plugins-8.0.31-1.el7.x86_64.rpm && \
test -f mysql-community-common-8.0.31-1.el7.x86_64.rpm && \
test -f mysql-community-icu-data-files-8.0.31-1.el7.x86_64.rpm && \
test -f mysql-community-libs-8.0.31-1.el7.x86_64.rpm && \
test -f mysql-community-libs-compat-8.0.31-1.el7.x86_64.rpm && \
test -f mysql-community-server-8.0.31-1.el7.x86_64.rpm || exit 1

# 卸载 MySQL
systemctl stop mysql mysqld 2>/dev/null
rpm -qa | grep -i 'mysql\|mariadb' | xargs -n1 rpm -e --nodeps
2>/dev/null
rm -rf /var/lib/mysql /var/log/mysqld.log /usr/lib64/mysql
/etc/my.cnf /usr/my.cnf
```

```

set -e
# 安装并启动 MySQL
yum install -y *.rpm >/dev/null 2>&1
systemctl start mysqld

#更改密码级别并重启 MySQL
sed -i
' /\[mysqld\]/avalidate_password.length=4\nvalidate_password.polic
y=0' /etc/my.cnf
systemctl restart mysqld

# 更改 MySQL 配置
tpass=$(cat /var/log/mysqld.log | grep "temporary password" | awk
'{print $NF}')
cat << EOF | mysql -uroot -p"${tpass}" --connect-expired-
password >/dev/null 2>&1
set password='000000';
update mysql.user set host='%' where user='root';
alter user 'root'@'%' identified with mysql_native_password by
'000000';
flush privileges;
EOF

```


执行脚本。

```
[root@hadoop102 mysql]# sh install_mysql.sh
```

#### 4) 退出 root 用户到 iflytek 用户

```
[root@hadoop102 mysql]# exit
```

### 3 连接 MySQL



MySQL - 新建连接

常规 高级 数据库 SSL SSH HTTP

Navicat 数据库

连接名: hadoop102

主机: hadoop102

端口: 3306

用户名: root

密码: ●●●●●●

☒ 保存密码

注意：如果报以下错误，通常是脚本执行问题，出现版本不兼容

```

报错: Client does not support authentication protocol requested by
server;consider upgrading Mysql client

```

解决办法如下:

(1) Hadoop102 进入 mysql

```
[iflytek@hadoop102 ~]$ mysql -uroot -p'000000'
```

(2) 执行如下 sql 语句并刷新

```
alter user 'root'@'%' identified with mysql_native_password by '000000';
```

```
flush privileges;
```

执行完毕, 重新测试 Navicat 中的连接

## 4 Mock 脚本使用

在/opt/module 中新建 mock-medical 目录, 将 mock-medical-1.1.jar 和 application.yaml 上传到该目录中:

```
[iflytek@hadoop102 ~]$ mkdir /opt/module/mock-medical
[iflytek@hadoop102 ~]$ cd /opt/module/mock-medical
# 上传文件
[iflytek@hadoop102 mock-medical]$ ll
总用量 34208
-rw-rw-r--. 1 iflytek iflytek      335 5月 12 08:47
application.yaml
-rw-rw-r--. 1 iflytek iflytek 35024769 5月 12 08:46 mock-medical-
1.1.jar
```

编辑 application.yaml, 配置数据连接:

```
spring:
  profiles:
    active: common
  datasource:
    url:
'jdbc:mysql://hadoop102:3306/medical?useSSL=false&allowPublicKeyR
etrieval=true'
    username: 'root'
    password: '000000'

mock:
  # 模拟数据生成的时间起始日期
  date: '2023-05-09'
  # 取值范围 0-1, 数值越大, 数据时间越紧凑
  enthusiasm: 0.1
```

首先在 MySQL 中创建数据库:

```
[iflytek@hadoop102 mock-medical]$ mysql -uroot -p000000 -e"drop
database if exists medical;create database medical charset
utf8mb4 collate utf8mb4_general_ci;"
```

之后按照命令行提示, 执行以下命令:

```
[iflytek@hadoop102 mock-medical]$ java -jar mock-medical-1.1.jar
```

即可生成 2024-07-09 日的数据。多次执行该 jar 包可以不断生成业务数据, 每执行一次

业务日期推后一天。

将以上的命令封装为一个脚本，以后方便处理：

```
[iflytek@hadoop102 ~]$ mkdir -p /home/iflytek/bin
[iflytek@hadoop102 ~]$ vim /home/iflytek/bin/medical_mock.sh
```

内容如下:

```
#!/bin/bash

for ((i=0; i < $1; i++))
do
    echo "正在执行第 $[ $i + 1 ] 次数据模拟"
    ssh hadoop102 "cd /opt/module/mock-medical/; java -jar mock-
medical-1.1.jar"
done
```

添加执行权限后。

```
[iflytek@hadoop102 ~]$ chmod +x /home/iflytek/bin/medical_mock.sh
```

执行 `medical_mock.sh` 即可生成指定天数的数据，用法如下。

```
[iflytek@hadoop102 bin]$ medical_mock.sh
正在执行第 1 次数据模拟
正在初始化 Spring Boot 环境
```

```
:: Spring Boot ::
(v2.7.11)
```

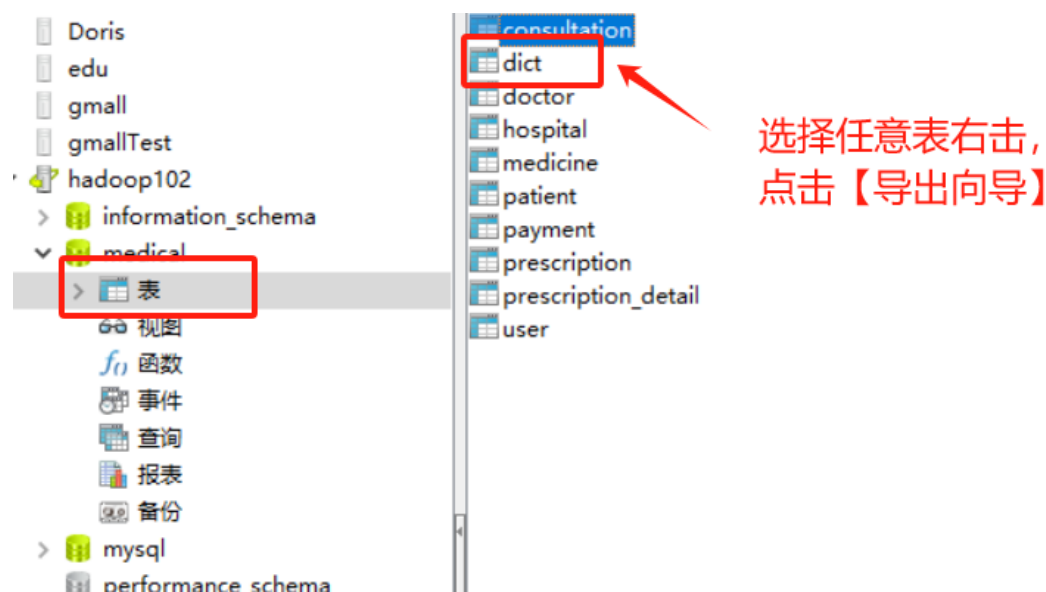
初始化数据中...

正在模拟 2024-08-01T16:59

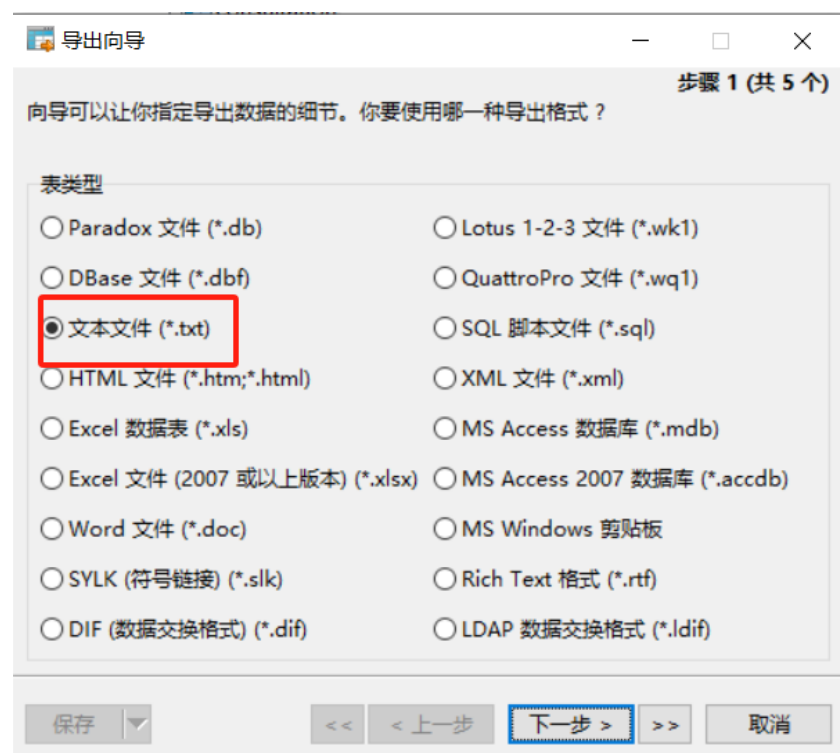
## 5 数据导出

**注意：数据导出之前需要至少生成 4-5 天的数据**

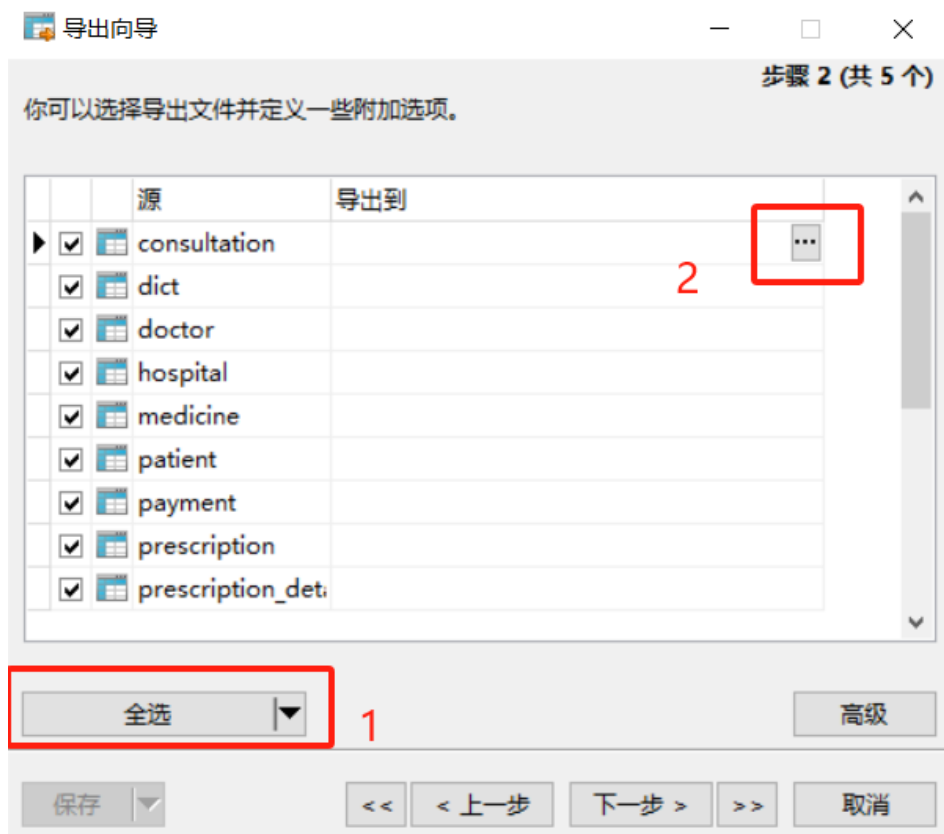
### (1) 第一步



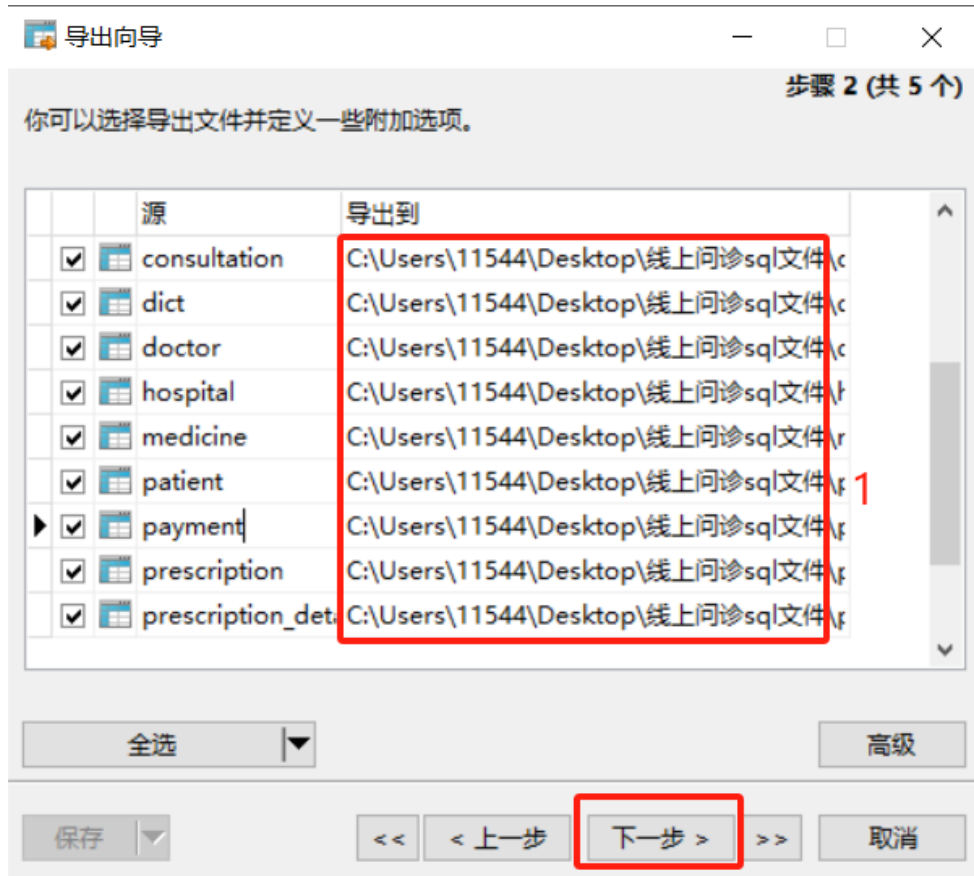
(2) 第二步 选择【文本文件】格式，点击【下一步】



(3) 第三步点击【全选】，为每一个数据源选择导出的目录，这里建议所有源文件放在同一个目录下



(4) 第四步，选择好所有地址，点击【下一步】



(5) 继续点击【下一步】

导出向导

步骤 3 (共 5 个)

你可以选择导出哪些列。

源表: consultation

可用栏位:

- ☒ id
- ☒ create\_time
- ☒ update\_time
- ☒ consultation\_fee
- ☒ description
- ☒ diagnosis
- ☒ rating
- ☒ review
- ☒ status
- ☒ doctor\_id
- ☒ patient\_id
- ☒ user\_id

☒ 全部栏位

(6) 第六步，分隔符选用【,】，方便后续统一格式建表，点击【下一步】

导出向导

步骤 4 (共 5 个)

你可以定义一些附加的选项。

☐ 包含列的标题

☐ 添加

☒ 遇到错误继续

行分隔符: CRLF

文本限定符: 无

栏位分隔符

- ☐ 定位
- ☐ 分号 (;)
- ☒ 逗号 (,)
- ☐ 空格
- ☐ 无
- ☐ 其他符号

日期、时间和数字

日期排序: YMD

日期分隔符: /

时间分隔符: :

小数点符号: .



(7) 点击【开始】，等待数据导出完毕

