SQL优化

测试目标

验证SQL优化的效果，了解索引对查询和数据插入的影响程度。

测试环境

1、jmeter模拟15个程序不停的执行SQL

```
select * from t_test where col_b = ?
```

其中“? ”为测试工具产生的1-100万随机数字，以此作为查询条件执行SQL
2、Oracle中有一张表t_test，预先插入了100万行记录，其中col_a是主键字段，内容为24位长的字符串；col_b是数值字段，内容为1到100万的随机数字（有重复值）；col_c为100位长的随机字符串；col_d为当前日前500天内的随机日期，日期类型。

测试方法

1、运行压力测试程序，查看在没有索引的情况下，每秒能执行完成几次SQL
2、为col_b字段，创建索引
3、再次运行压力测试程序，查看在有索引的情况下，每秒能执行完成几次SQL
4、比较两次结果，验证SQL优化效果

oracle安装

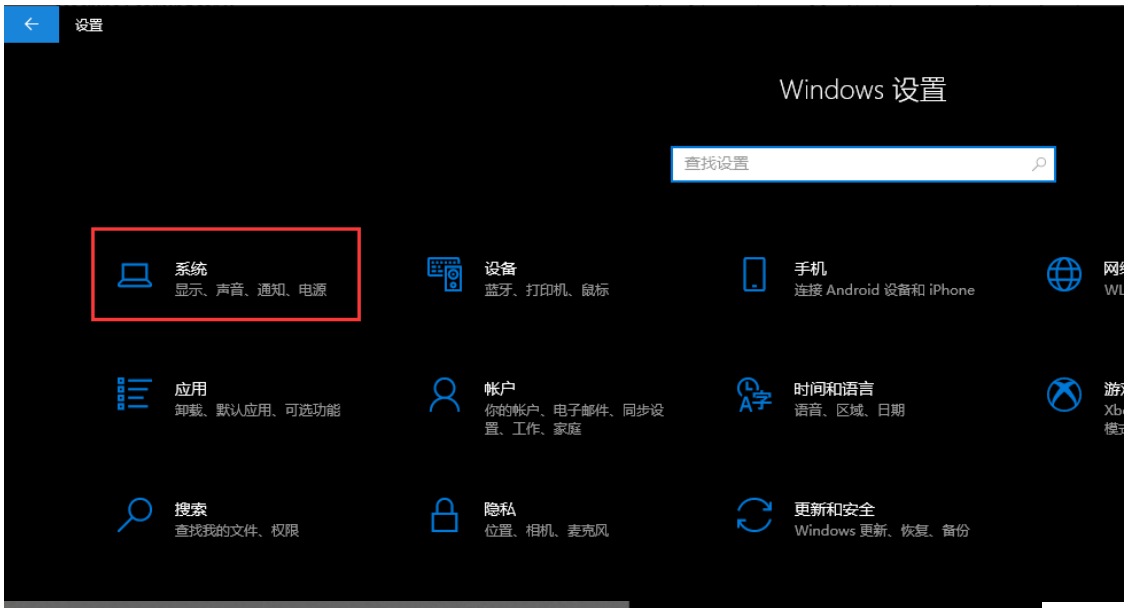
通过docker安装oracle-xe

```
docker run -d --name oracle-xe -p 49161:1521 -e ORACLE_ALLOW_REMOTE=true
oracleinanutshell/oracle-xe-11g
```

安装Oracle instant client

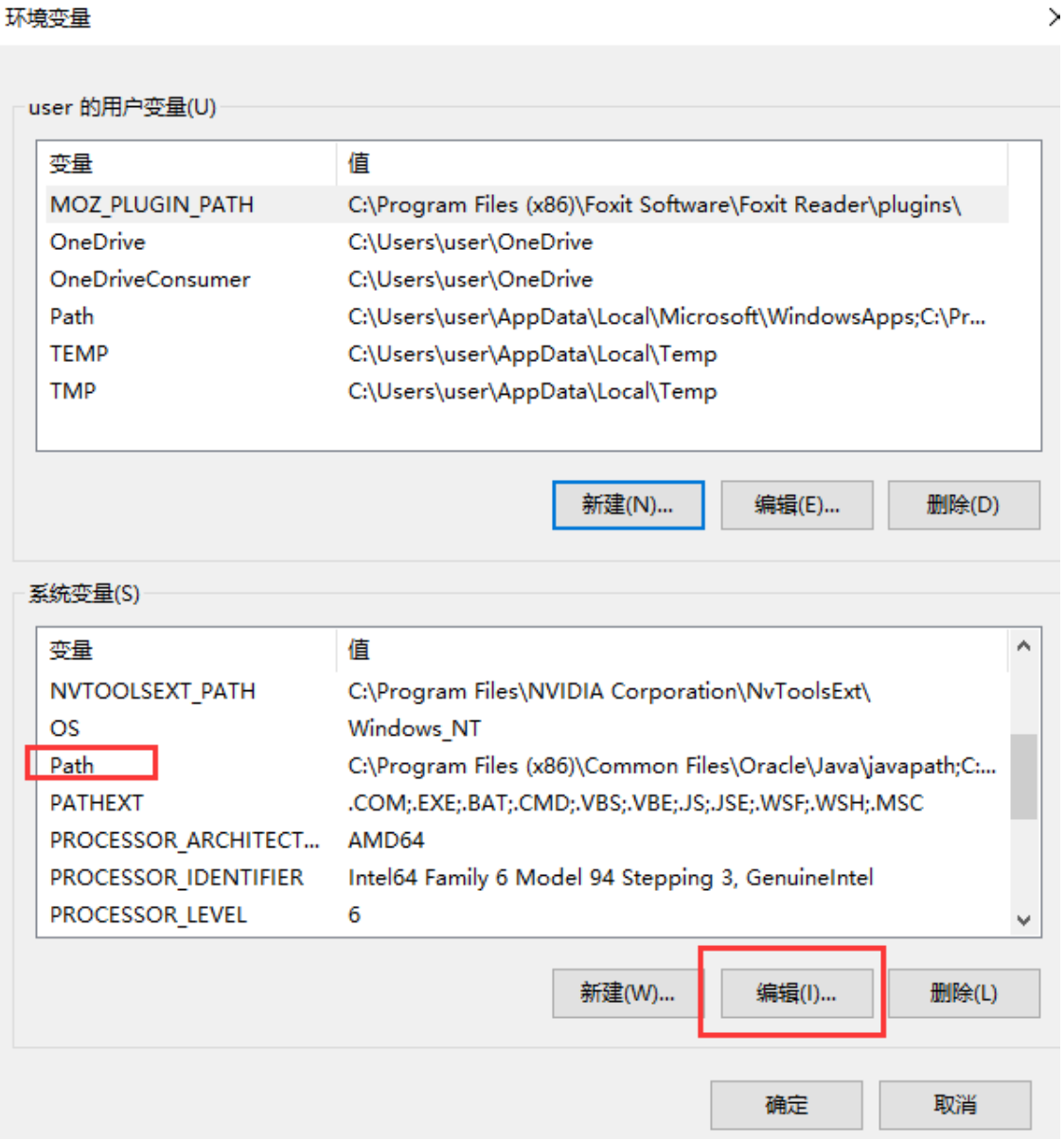
1. 从Oracle官方网站下载Basic Package和SQL*Plus Package，解压缩放到一个目录下（例如：c:\oracle_instant_client）。

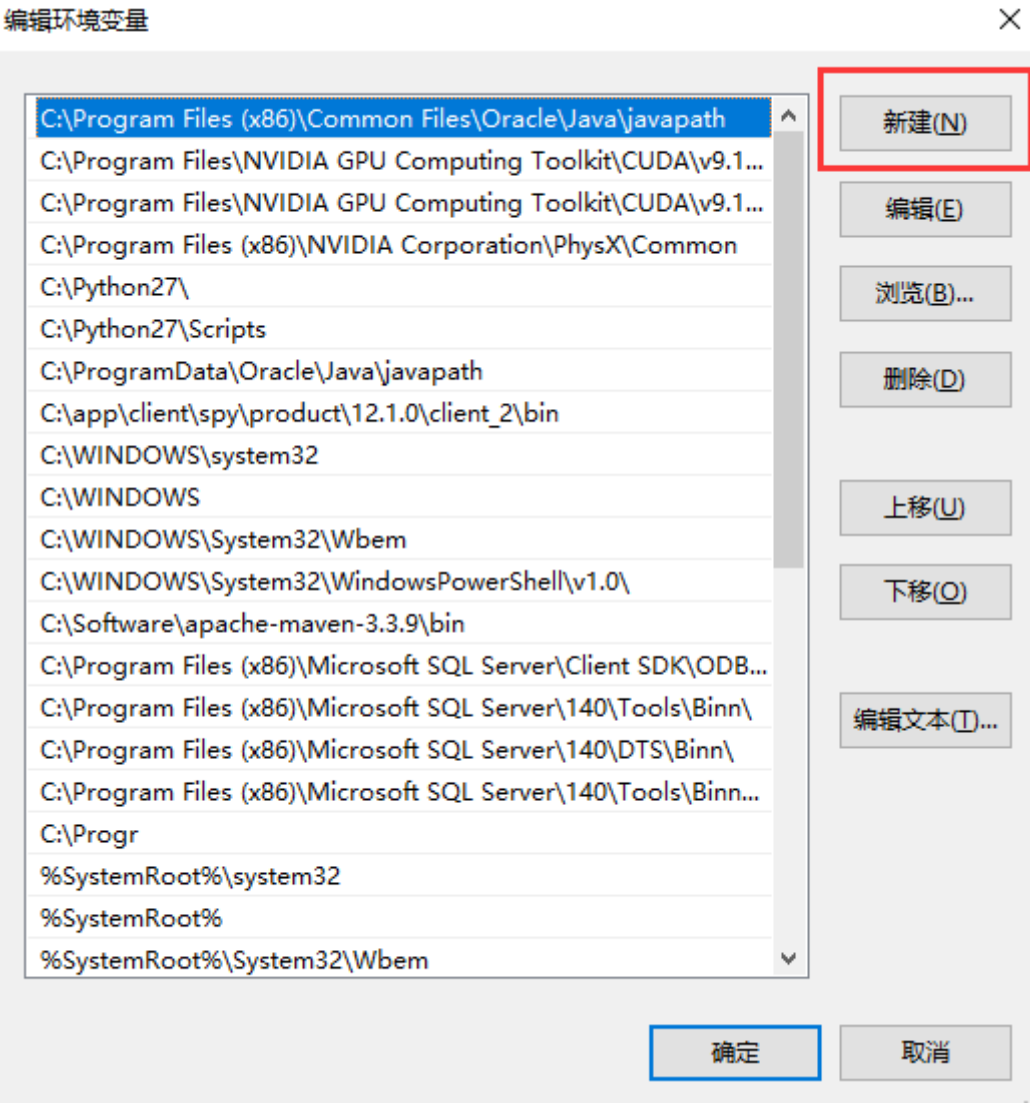
2. 设置系统环境变量


设置
Windows 设置
查找设置
系统
显示、声音、通知、电源
设备
蓝牙、打印机、鼠标
手机
连接 Android 设备和 iPhone
应用
卸载、默认应用、可选功能
帐户
你的帐户、电子邮件、同步设置、工作、家庭
时间和语言
语音、区域、日期
搜索
查找我的文件、权限
隐私
位置、相机、麦克风
更新和安全
Windows 更新、恢复、备份


设置
主页
查找设置
系统
通知和操作
专注助手
电源和睡眠
存储
平板电脑
多任务处理
投影到此电脑
体验共享
剪贴板
远程桌面
关于
显示
重新排列显示器
选择下面的某一显示器以更改
颜色
夜间模式(19:28 之前保持关闭)
开
夜间模式设置
颜色配置文件

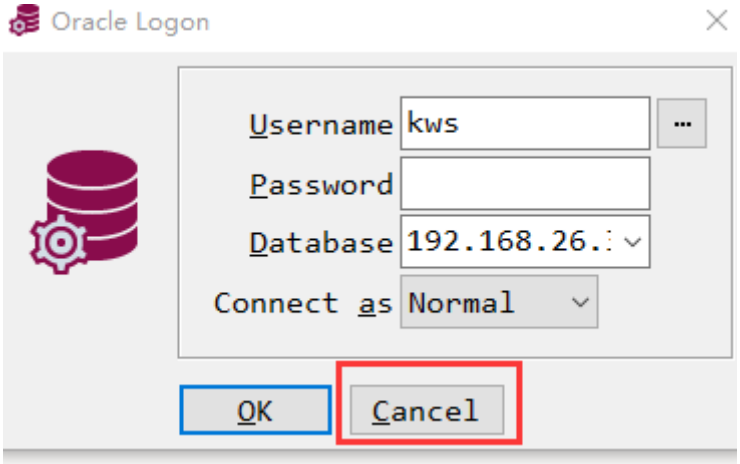
修改path环境变量，加入oracle instant client路径


环境变量
user 的用户变量(U)
变量
值
MOZ_PLUGIN_PATH
C:\Program Files (x86)\Foxit Software\Foxit Reader\plugins\
OneDrive
C:\Users\user\OneDrive
OneDriveConsumer
C:\Users\user\OneDrive
Path
C:\Users\user\AppData\Local\Microsoft\WindowsApps;C:\Pr...
TEMP
C:\Users\user\AppData\Local\Temp
TMP
C:\Users\user\AppData\Local\Temp
新建(N)...
编辑(E)...
删除(D)
系统变量(S)
变量
值
NVTOOLSEXT_PATH
C:\Program Files\NVIDIA Corporation\NvToolsExt\
OS
Windows_NT
Path
C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:...
PATHEXT
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECT...
AMD64
PROCESSOR_IDENTIFIER
Intel64 Family 6 Model 94 Stepping 3, GenuineIntel
PROCESSOR_LEVEL
6
新建(W)...
编辑(I)...
删除(L)
确定
取消

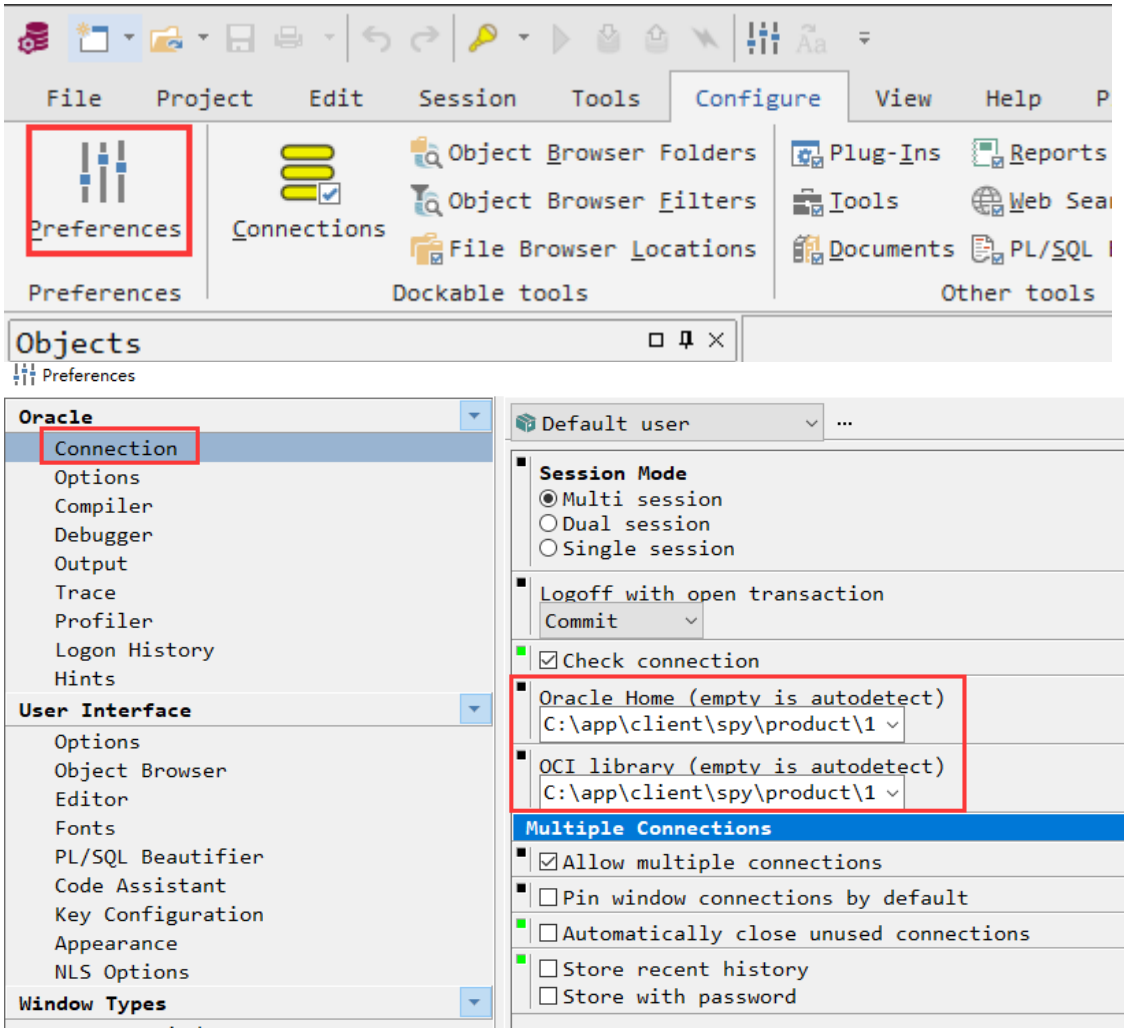


通过pl/sql developer连接到oracle-xe

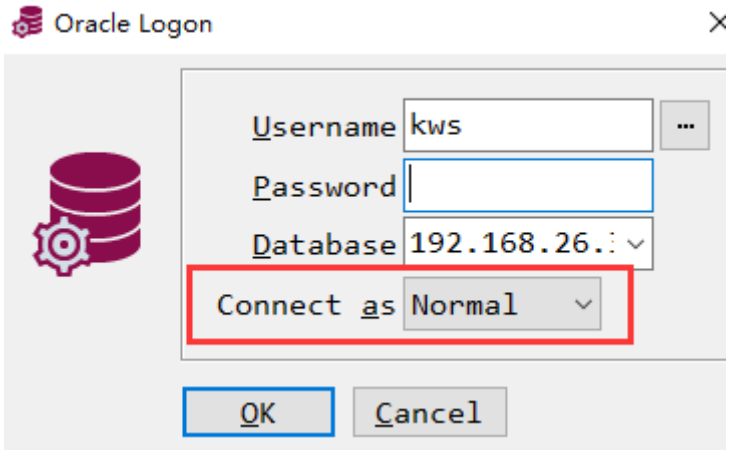
1. 设置oracle DLL文件路径
   第一次打开程序，需要点击cancel，进入系统。



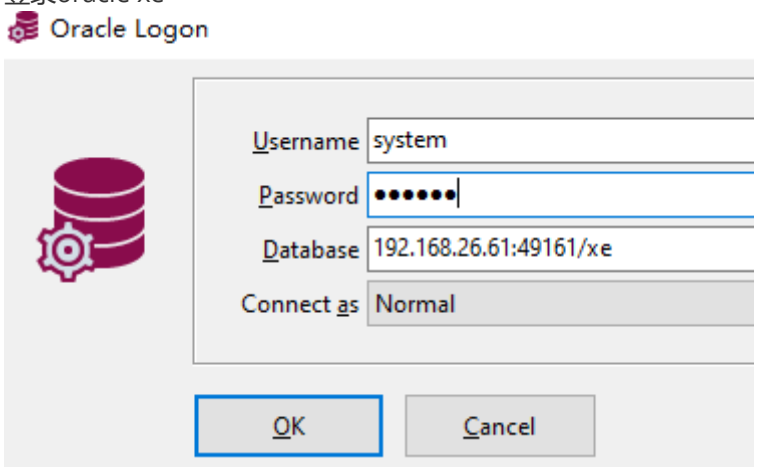
打开参数页



设置完毕以后，如果再次打开程序，会出现红框内容。如果没有，那么路径设置，可能存在问题。



2. 登录oracle xe



username: system

password: oracle

database: [ip]:49161/xe

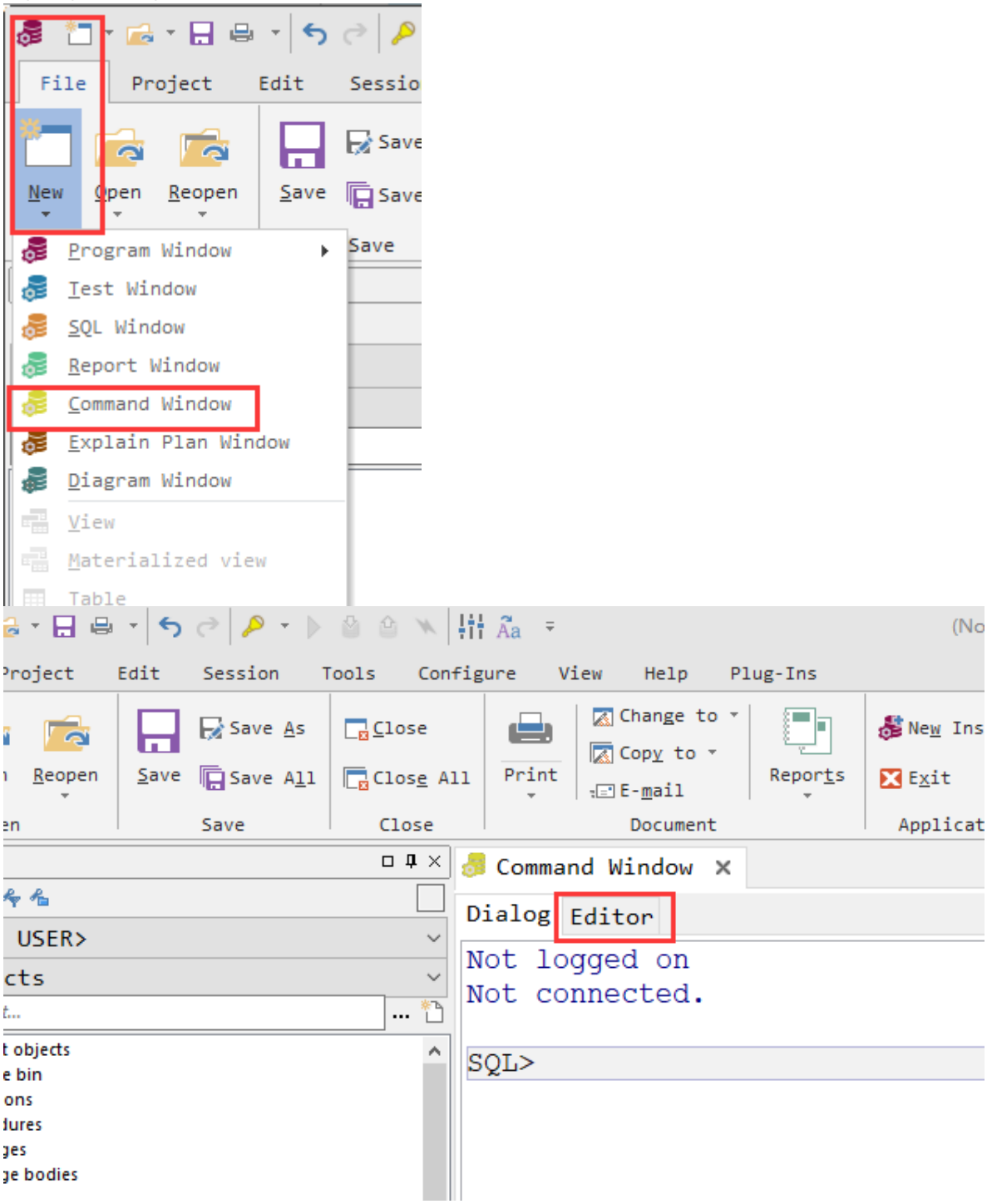
这里的IP，指的是wsl的IP地址。可以进入wsl，通过执行命令获取

```
root@CHENZHEN-NUC:/home/spy/jmeter-docker/docker-jmeter/tests/trivial#
root@CHENZHEN-NUC:/home/spy/jmeter-docker/docker-jmeter/tests/trivial# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: bond0: <BROADCAST,MULTICAST,MASTER> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether fa:66:6b:d3:56:cd brd ff:ff:ff:ff:ff:ff
3: dummy0: <BROADCAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether fa:7c:6e:e1:10:f6 brd ff:ff:ff:ff:ff:ff
4: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000
    link/sit 0.0.0.0 brd 0.0.0.0
5: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:b8:3d:03 brd ff:ff:ff:ff:ff:ff
    inet 172.21.127.178/20 brd 172.21.127.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:feb8:3d03/64 scope link
       valid_lft forever preferred_lft forever
6: docker0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:f9:a9:13:12 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
```

搭建性能测试环境

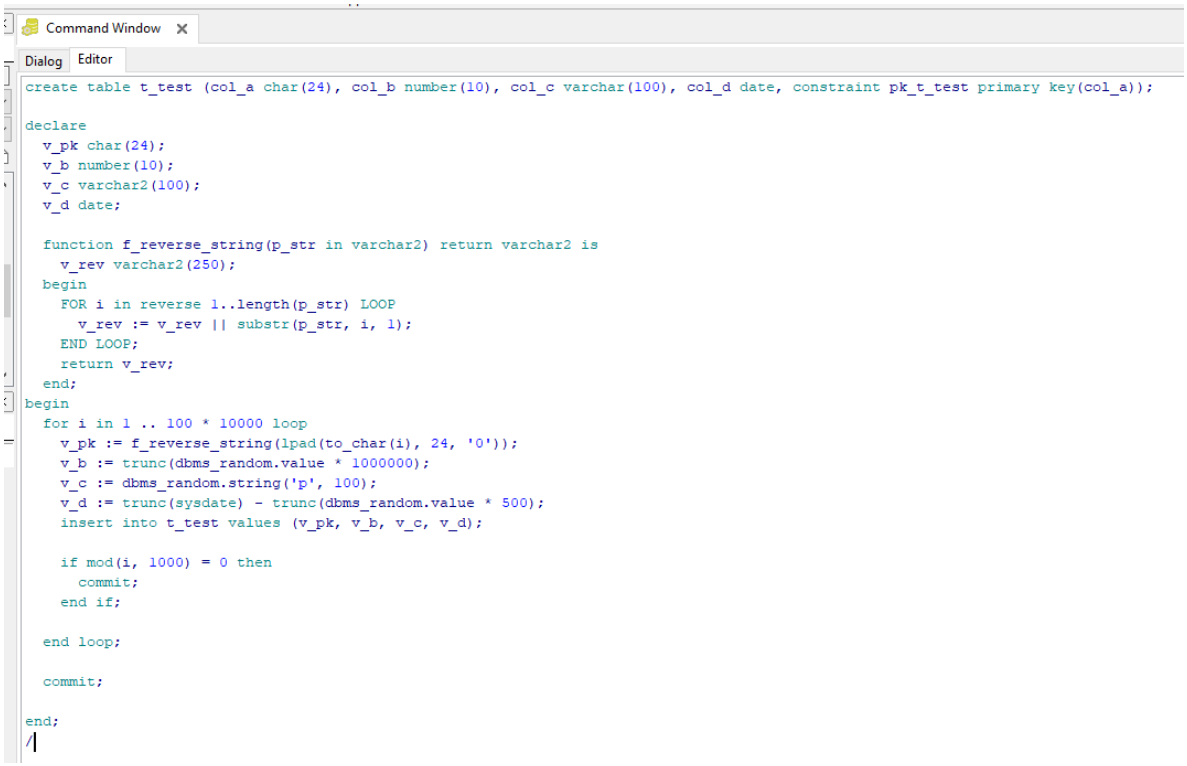
在oracle中灌入测试数据

在pl/sql developer中点击New，然后选择command window，然后选择Editor页，粘贴入以下SQL

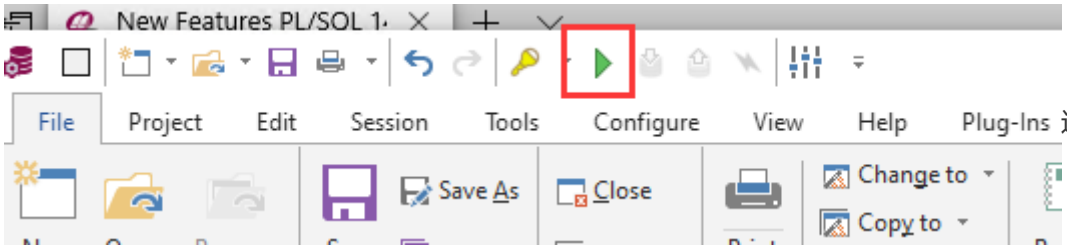



```
create table t_test (col_a char(24), col_b number(10), col_c varchar(100), col_d
date, constraint pk_t_test primary key(col_a));

declare
  v_pk char(24);
  v_b number(10);
  v_c varchar2(100);
  v_d date;

  function f_reverse_string(p_str in varchar2) return varchar2 is
    v_rev varchar2(250);
  begin
    FOR i in reverse 1..length(p_str) LOOP
      v_rev := v_rev || substr(p_str, i, 1);
    END LOOP;
```

```
    return v_rev;
  end;
begin
  for i in 1 .. 100 * 10000 loop
    v_pk := f_reverse_string(lpad(to_char(i), 24, '0'));
    v_b := trunc(dbms_random.value * 1000000);
    v_c := dbms_random.string('p', 100);
    v_d := trunc(sysdate) - trunc(dbms_random.value * 500);
    insert into t_test values (v_pk, v_b, v_c, v_d);

    if mod(i, 1000) = 0 then
      commit;
    end if;

  end loop;

  commit;

end;
/
```



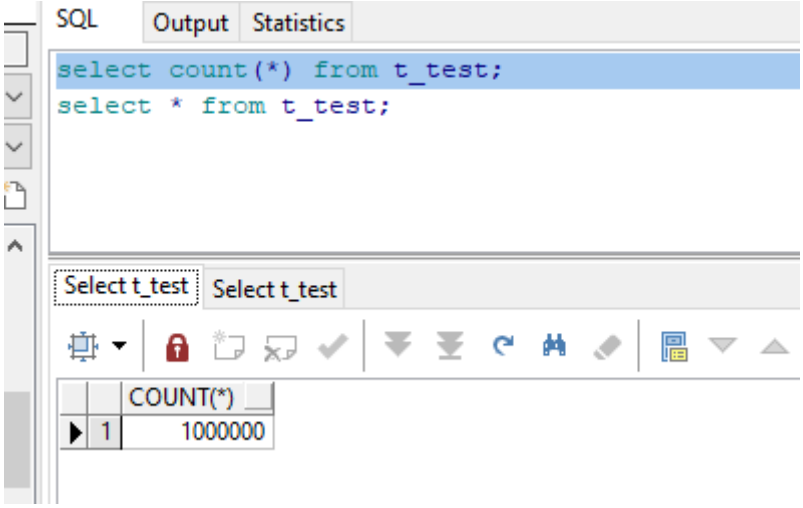
然后点击执行按钮



运行SQL语句，生成测试数据

点击new --> SQL window，打开一个SQL执行窗口，输入以下SQL，检查执行结果

```
select count(*) from t_test;
```



安装并配置jmeter

1. 下载jmeter

```
wget https://mirror.bit.edu.cn/apache//jmeter/binaries/apache-jmeter-5.3.tgz
tar -xvf apache-jmeter-5.3.tgz
sudo apt install openjdk-8-jdk unzip
```

这里运行apt时候，可能系统会提示你输入wsl的用户密码

2. 检测安装是否成功

```
cd apache-jmeter-5.3
./bin/jmeter -v
```

出现以下内容表示安装成功

```
spy@CHENZHEN-NUC:~/jmeter-docker/apache-jmeter-5.3$ ./bin/jmeter -v
    _    ____   _    ____ _   _ _____       _ __  __ _____ _____ _____ ____
   / \  |  _ \ / \  / ___| | | | ____|     | |  \/  | ____|_   _| ____|  _ \
  / _ \ | |_) / _ \| |   | |_| |  _|    _  | | |\/| |  _|   | | |  _| | |_) |
 / ___ \|  __/ ___ \ |___|  _  | |___  | |_| | |  | | |___  | | | |___|  _ <
/_/   \_\_| /_/   \_\____|_| |_|_____|  \___/|_|  |_|_____| |_| |_____|_| \_\ 5.3

Copyright (c) 1999-2020 The Apache Software Foundation
```

3. 配置jmeter环境

```
cp ~/bjppt/files/ojdbc7.jar ~/apache-jmeter-5.3/lib/
mkdir ~/apache-jmeter-5.3/tests
cp ~/bjppt/files/test_plan.jmx ~/apache-jmeter-5.3/tests/
cp ~/bjppt/files/jmeter-jdbc-test.sh ~/apache-jmeter-5.3/
```

运行性能测试，验证SQL优化效果

运行第一次性能测试，持续3分钟后，中断测试，观察并记录结果

```
cd ~/apache-jmeter-5.3
./jmeter-jdbc-test.sh
```

```
spy@CHENZHEN-NUC:~/apache-jmeter-5.3$ ./jmeter-jdbc-test.sh
Creating summariser <summary>
Created the tree successfully using tests/test_plan.jmx
Starting standalone test @ Fri May 22 16:42:01 CST 2020 (1590136921085)
Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445
summary +      1 in 00:00:01 =    0.9/s Avg:  1049 Min:  1049 Max:  1049 Err:     0 (0.00%) Active: 15 Started: 15 Finished: 0
summary +   1075 in 00:00:27 =   39.2/s Avg:   386 Min:    35 Max:  2447 Err:     0 (0.00%) Active: 15 Started: 15 Finished: 0
summary =   1076 in 00:00:29 =   37.6/s Avg:   387 Min:    35 Max:  2447 Err:     0 (0.00%)
summary +   1177 in 00:00:30 =   39.3/s Avg:   381 Min:    34 Max:  2001 Err:     0 (0.00%) Active: 15 Started: 15 Finished: 0
summary =   2253 in 00:00:59 =   38.5/s Avg:   384 Min:    34 Max:  2447 Err:     0 (0.00%)
summary +   1157 in 00:00:30 =   38.6/s Avg:   388 Min:    36 Max:  2026 Err:     0 (0.00%) Active: 15 Started: 15 Finished: 0
summary =   3410 in 00:01:29 =   38.5/s Avg:   385 Min:    34 Max:  2447 Err:     0 (0.00%)
summary +   1208 in 00:00:30 =   40.3/s Avg:   372 Min:    35 Max:  2209 Err:     0 (0.00%) Active: 15 Started: 15 Finished: 0
summary =   4618 in 00:01:59 =   38.9/s Avg:   382 Min:    34 Max:  2447 Err:     0 (0.00%)
```

图中，红色每隔30秒，会出现2行，带+的行表示30s内的测试结果，带+的行表示累计测试结果。黄色的内容为每秒执行的数量。
这里的图上内容表示：在00:00:30到00:00:59这段时间内，平均每秒有39.3条SQL执行完成，而从本次测试累计每秒有38.5个SQL执行完成

在oracle中，对表增加索引

在pl/sql developer中，新建sql window，然后执行以下SQL

```
create index idx_t_test_colb on t_test(col_b);
```

运行第二次性能测试，持续3分钟后，中断测试，观察并记录结果

```
cd ~/apache-jmeter-5.3
./jmeter-jdbc-test.sh
```

```
spy@CHENZHEN-NUC:~/apache-jmeter-5.3$ cd ~/apache-jmeter-5.3
spy@CHENZHEN-NUC:~/apache-jmeter-5.3$ ./jmeter-jdbc-test.sh
Creating summariser <summary>
Created the tree successfully using tests/test_plan.jmx
Starting standalone test @ Fri May 22 16:40:09 CST 2020 (1590136809621)
Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445
summary + 110128 in 00:00:20 = 5502.3/s Avg:     2 Min:     0 Max:   744 Err:     0 (0.00%) Active: 15 Started: 15 Finished: 0
summary + 202272 in 00:00:30 = 6742.4/s Avg:     2 Min:     0 Max:    51 Err:     0 (0.00%) Active: 15 Started: 15 Finished: 0
summary = 312400 in 00:00:50 = 6246.1/s Avg:     2 Min:     0 Max:   744 Err:     0 (0.00%)
summary + 196049 in 00:00:30 = 6535.0/s Avg:     2 Min:     0 Max:    38 Err:     0 (0.00%) Active: 15 Started: 15 Finished: 0
```

这里测试结果，可以看到每秒有6000左右的SQL执行完成。

测试结论

可以看到，数据库在增加索引前后的SQL查询性能差别巨大，从40增加到6000，大约提升150倍。