关于shell脚本执行Hive sql的若干问题
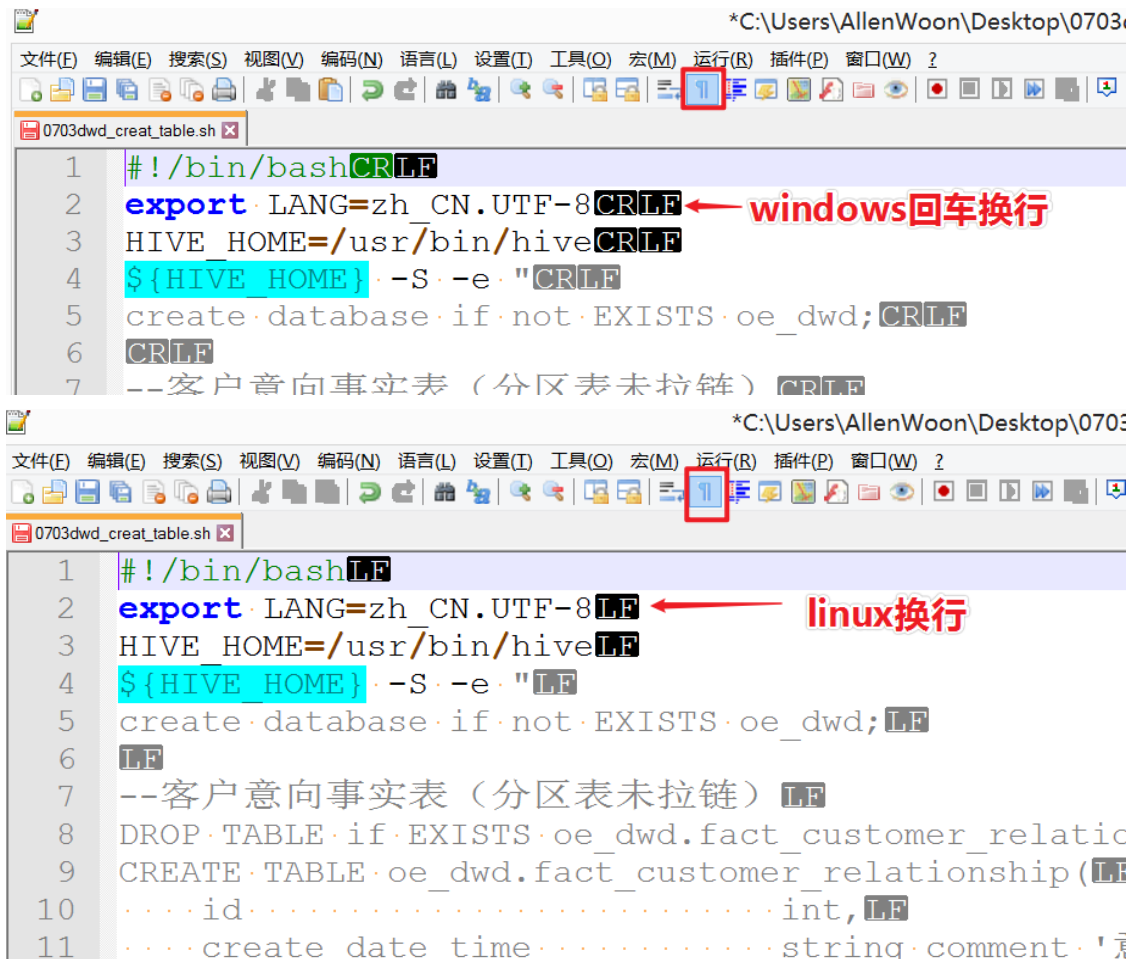
问题1：windows平台和linux平台回车换行符不一样

- 如果是在windows平台编写的shell脚本，直接上传到linux执行可能会出错。
- 原因是 windows平台换行符是/r/n　linux平台是/n

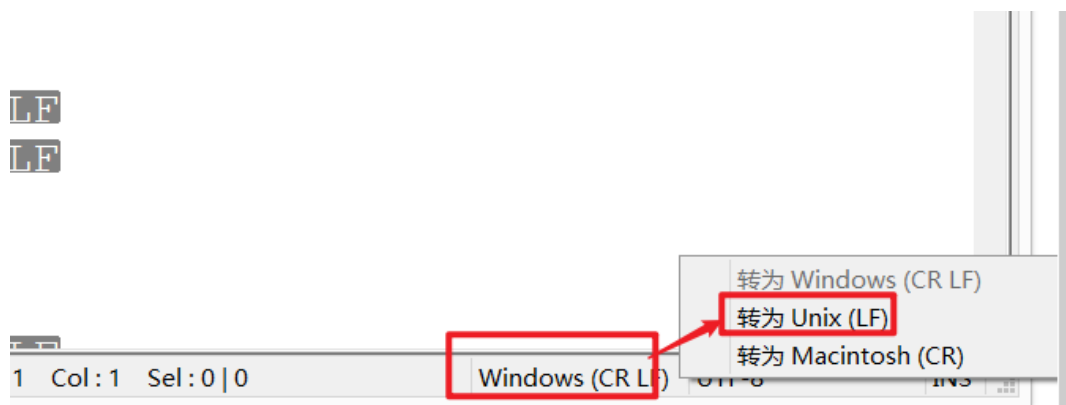

- 解决方式
  - 1、在linux上使用vim编辑器编写脚本；
  - 2、在windows平台编写好脚本 使用notepad++编辑器进行修改

    右下角 右键点击 选择unix

问题2：sql中``反引号和shell语言中反引号冲突问题

- 在sql语法中，如果遇到sql关键字 ，需要使用`反引号

- 而在shell语法中，`反引号表示执行里面的命令 因此就产生了冲突

- 举个栗子 hive建表sql如下：

```bash
#!/bin/bash
export LANG=zh_CN.UTF-8
HIVE_HOME=/usr/bin/hive
${HIVE_HOME} -S -e "
CREATE TABLE oe_dwd.fact_customer_clue(
    id                      int,
    create_date_time        string comment '创建时间',
    `user`                  string comment '所属坐席'
);"
```

其中user是hive sql关键字，需要使用反引号，但是这样执行会报如下的错误：

```
[root@hadoop01 ~]# sh 0703dwd_creat_table.sh
0703dwd_creat_table.sh:行4: user: 未找到命令
WARNING: Use "yarn jar" to launch YARN applications.
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/cloudera/parcels/
-2.8.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/cloudera/parcels/
```

因为shell脚本又把`反引号理解成了执行里面的命令，即把user当成的一个命令如执行

明细shell没有这样的命令，所以执行报错 找不到命令。

解决方案： **使用\将`反引号进行转义**，如下所示

```bash
#!/bin/bash
export LANG=zh_CN.UTF-8
HIVE_HOME=/usr/bin/hive
${HIVE_HOME} -S -e "
CREATE TABLE oe_dwd.fact_customer_clue(
    id                      int,
    create_date_time        string comment '创建时间',
    \`user\`                 string comment '所属坐席');"
```


问题3：shell的执行方式

在linux的/root下有一个shell脚本，如何去执行的呢？

```
[root@hadoop01 ~]# pwd
/root
[root@hadoop01 ~]# ll
总用量 4
-rw-r--r-- 1 root root 29 7月   3 18:52 hello.sh
[root@hadoop01 ~]#
```

- 相对路径

```
[root@hadoop01 ~]# ll
总用量 4
-rw-r--r-- 1 root root 29 7月    3 18:52 hello.sh
[root@hadoop01 ~]# chmod u+x hello.sh
[root@hadoop01 ~]# ./hello.sh
hello shell
[root@hadoop01 ~]#
```

- <mark>绝对路径</mark>

```
hello shell
[root@hadoop01 ~]# /root/hello.sh
hello shell
[root@hadoop01 ~]#
```

- <mark>sh+shell</mark>

```
hello shell
[root@hadoop01 ~]# sh hello.sh
hello shell
[root@hadoop01 ~]# sh /root/hello.sh
hello shell
[root@hadoop01 ~]#
```