

synchronizedb.sh

```
#!/bin/bash

#author by:qiankun last_edit:2016-07-21

set -x

DATETIME=`date +%Y%m%d %H:%M:%S` #running time
DAY=`date +%Y%m%d`
MYSQL=`which mysql`
MYSQLDUMP=`which mysqldump`
USER="qiankun"
PASSWD="WYaG6a#s9aOE"
HOST="172.16.88.243"
PORT_LOCAL=3316
HOST2="172.16.88.240"
PORT_REMOTE=3306
LOGNAME="/var/log/mysql_rsync.log"
INPUT="$1"

skipdb()
{
    cat /etc/rc.d/skip_table.txt | while read line
    do
        db=`echo $line | awk '{print $1}'`
        table=`echo $line | awk '{print $2}'`
        if [ $1 == 1 ]
        then
            $MYSQL -u$USER -p$PASSWD -P$PORT_LOCAL $db -e "alter table \"$db\".\"$table\" rename to \"$db\".\"$table\"_\"$DAY\"_tmp;"
        else
            $MYSQL -u$USER -p$PASSWD -P$PORT_LOCAL $db -e "drop table \"$db\".\"$table\";alter table \"$db\".\"$table\"_\"$DAY\"_tmp rename to \"$db\".\"$table\";"
        fi
    done
}

allsync()
{
    echo "#####" >> $LOGNAME
    echo "$DATETIME" >> $LOGNAME
    $MYSQL -u$USER -p$PASSWD -P$PORT_REMOTE -h$HOST -NBe "stop slave sql_thread;"
    for i in `cat /etc/rc.d/dblist_final.txt`
    do
        $MYSQL -u$USER -p$PASSWD -P$PORT_REMOTE -h$HOST -NBe "show create database $i;"
        if [ $? != 0 ];then
            continue
        fi
        $MYSQL -u$USER -p$PASSWD -P$PORT_LOCAL -NBe "show create database $i;"
        if [ $? != 0 ];then
            $MYSQL -u$USER -p$PASSWD -P$PORT_LOCAL -NBe "create database $i;"
        fi
        CHARACTER=`$MYSQL -u$USER -p$PASSWD -P$PORT_REMOTE -h$HOST -NBe "SELECT DEFAULT_CHARACTER_SET_NAME FROM
information_schema.SCHEMATA WHERE SCHEMA_NAME='$i';" 2>/dev/null`
        $MYSQLDUMP -u$USER -p$PASSWD -P$PORT_REMOTE -h$HOST --quick --triggers --routines --default-character-set=$CHARACTER --
single-transaction --max_allowed_packet=512M $i | $MYSQL -u$USER -p$PASSWD -P$PORT_LOCAL --default-character-set=$CHARACTER $i
        #echo $i
        if [ $? == 0 ];then
```

```

        echo "$i success" >> $LOGNAME
    else
        echo "$i fail" >> $LOGNAME
    fi

done

$MYSQL -u$USER -p$PASSWD -P$PORT_LOCAL wms_inventory < /etc/rc.d/views_4.10.sql
$MYSQL -u$USER -p$PASSWD -P$PORT_REMOTE -h$HOST -NBe "start slave sql_thread;"
echo "#####" >> $LOGNAME
echo "" >> $LOGNAME
IRONMANsync
}

```

##\$? : 显示最后命令的退出状态, 0表示没有错误, 其他表示有错误

```

employeesync()
{
    TABLES=(admin admin_department admin_function department employee function)
    $MYSQLDUMP -u$USER -p$PASSWD -P$PORT_REMOTE -h$HOST --single-transaction employee --tables ${TABLES[@]} | $MYSQL -u$USER -
p$PASSWD -P$PORT_LOCAL employee 2>/dev/null
    $MYSQL -u$USER -p$PASSWD -P$PORT_LOCAL employee -NBe "update employee set password = '96e79218965eb72c92a549dd5a330112';"
    if [ $? == 0 ];then
        echo "update employee success" >> $LOGNAME
    else
        echo "update employee fail" >> $LOGNAME
    fi
    echo "" >> $LOGNAME
}

```

#2>/dev/null : 将报错信息丢弃

```

IRONMANsync()
{
    $MYSQLDUMP -u$USER -p$PASSWD -P$PORT_REMOTE -h$HOST2 --single-transaction IRONMAN_v2 | $MYSQL -u$USER -p$PASSWD -P$PORT_LOCAL
IRONMAN_v2 2>/dev/null
    if [ $? == 0 ];then
        echo "IRONMAN_v2 success" >> $LOGNAME
    else
        echo "IRONMAN_v2 fail" >> $LOGNAME
    fi
    echo "" >> $LOGNAME
}

```

```

singlesync()
{
    echo "$DATETIME" >> $LOGNAME
    read -p "请输入你要迁移的数据库名称(限1个): " sname
    if [ $sname == "IRONMAN_v2" ];then
        IRONMANsync
    else
        $MYSQL -u$USER -p$PASSWD -P$PORT_REMOTE -h$HOST -NBe "show create database $sname;" > /dev/null 2>&1
        if [ $? != 0 ];then
            echo "没有该数据库, 请重新确认!"
            exit 1
        fi
        $MYSQL -u$USER -p$PASSWD -P$PORT_LOCAL -NBe "show create database $sname;" > /dev/null 2>&1
        if [ $? != 0 ];then
            echo "不存在该数据库, 已在预发布创建!"

```

```

        $MYSQL -u$USER -p$PASSWD -P$PORT_LOCAL -NBe "create database $sname;" 2>/dev/null
    fi
    CHARACTER=`$MYSQL -u$USER -p$PASSWD -P$PORT_REMOTE -h$HOST -BNe "SELECT DEFAULT_CHARACTER_SET_NAME FROM
information_schema.SCHEMATA WHERE SCHEMA_NAME=' $sname';"` 2>/dev/null
    $MYSQLDUMP -u$USER -p$PASSWD -P$PORT_REMOTE -h$HOST $OPTION --quick --triggers --routines --default-character-set=$CHARACTER
--single-transaction --max_allowed_packet=512M $sname | $MYSQL -u$USER -p$PASSWD -P$PORT_LOCAL --default-character-set=$CHARACTER
$sname 2>/dev/null
    #echo $sname
    if [ $? == 0 ];then
        echo "$sname success" >> $LOGNAME
    else
        echo "$sname fail" >> $LOGNAME
    fi
    echo "" >> $LOGNAME
fi
}

special()
{
    echo "$DATETIME" >> $LOGNAME
    read -p "请按照下列格式输入你要迁移的数据库名和表名(qiankun.Seconds_Behind qiankun.Position_Check):" sptable
    #echo $sptable
    for m in $sptable
    do
        SPTABLE=`echo $m | awk -F ' ' '{print $2}'`
        SPDB=`echo $m | awk -F ' ' '{print $1}'`
        if [ ! $SPTABLE ] || [ ! $SPDB ];then
            echo "$m的输入的格式不正确, 终止运行!"
            exit 1
        fi
        $MYSQL -u$USER -p$PASSWD -P$PORT_REMOTE -h$HOST -NBe "show create table $m;" > /dev/null 2>&1
        if [ $? != 0 ];then
            echo "没有数据表$m, 跳过该表!"
            continue
        fi
        $MYSQLDUMP -u$USER -p$PASSWD -P$PORT_REMOTE -h$HOST $OPTION --single-transaction --max_allowed_packet=512M $SPDB $SPTABLE
| $MYSQL -u$USER -p$PASSWD -P$PORT_LOCAL $SPDB 2>/dev/null
        if [ $? == 0 ];then
            echo "$M success" >> $LOGNAME
        else
            echo "$M fail" >> $LOGNAME
        fi
        echo "" >> $LOGNAME
    done
}

multisync()
{
    echo "$DATETIME" >> $LOGNAME
    read -p "请输入你需要迁移的所有库名(可多个), 名称之间用空格隔开即可:" $snames
    #echo $snames
    for m in $snames
    do
        echo $m
        $MYSQL -u$USER -p$PASSWD -P$PORT_REMOTE -h$HOST -NBe "show create database $m;" 2>/dev/null
        if [ $? != 0 ];then
            continue

```

```

fi
$MYSQL -u$USER -p$PASSWD -P$PORT_LOCAL -NBe "show create database $m;" 2>/dev/null
if [ $? != 0 ];then
    $MYSQL -u$USER -p$PASSWD -P$PORT_LOCAL -NBe "create database $i;" 2>/dev/null
fi
CHARACTER=`$MYSQL -u$USER -p$PASSWD -P$PORT_REMOTE -h$HOST -BNe "SELECT DEFAULT_CHARACTER_SET_NAME FROM
information_schema.SCHEMATA WHERE SCHEMA_NAME='$m';" 2>/dev/null`
$MYSQLDUMP -u$USER -p$PASSWD -P$PORT_REMOTE -h$HOST $OPTION --quick --triggers --routines --default-character-
set=$CHARACTER --single-transaction --max_allowed_packet=512M $m | $MYSQL -u$USER -p$PASSWD -P$PORT_LOCAL --default-character-
set=$CHARACTER $m 2>/dev/null
if [ $? == 0 ];then
    echo "$m success" >> $LOGNAME
else
    echo "$m fail" >> $LOGNAME
fi
done
echo "" >> $LOGNAME
}

if [ -z $INPUT ];then
    echo "请在脚本后面间隔一个空格加上下面3个参数中的一个:all single multi employeeupdate special"
    exit 1
elif [ $INPUT = "all" -o $INPUT = "whole" ];then
    if [ $INPUT = "all" ];then
        read -p "确认要导出所有数据到预发布环境?该操作会覆盖和更新当前所有数据! [yes/no]: " choice
        if [ $choice == "yes" -o $choice == "y" ]
        then
            allsync
            employeesync
        else
            echo "本次操作取消!"
            exit 1
        fi
    else
        skipdb 1
        allsync
        skipdb 2
        employeesync
    fi
elif [ $INPUT = "single" ];then
    read -p "请选择传输类型[yes(包含数据)/no(不包含数据)]: " optype
    if [ $optype == "yes" -o $optype == "y" ]
    then
        OPTION=""
    else
        OPTION="-d"
    fi
    singlesync
elif [ $INPUT = "multi" ];then
    read -p "选择传输类型[yes(包含数据)/no(不包含数据)]: " optype
    if [ $optype == "yes" -o $optype == "y" ]
    then
        OPTION=""
    else
        OPTION="-d"
    fi
    multisync
elif [ $INPUT = "employeesync" ];then

```

```
read -p "确认要手动同步线上employee的部分数据? [yes/no]: " optype
if [ $optype == "yes" -o $optype == "y" ];then
    employeesync
else
    continue
fi
elif [ $INPUT = "special" ];then
    read -p "选择传输类型[yes(包含数据)/no(不包含数据)]: " optype
    if [ $optype == "yes" -o $optype == "y" ]
    then
        OPTION=""
    else
        OPTION="-d"
    fi
    special
else
    echo "你输入的参数不正确!"
    exit 1
fi
```